Corrosion Segmentation on Steel Structures via Illumination-Aware Preprocessing and Bayesian U²-Net

A thesis submitted in fulfilment of the requirements

for the degree of Master of Engineering

Yangxiao Zheng

Supervised by

Dr Arthur Fang and Prof James Lim

School of Engineering

The University of Waikato

New Zealand

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2025

Abstract

Steel corrosion poses a major threat to infrastructure safety and longevity, demanding reliable and accurate automated inspection systems. This study presents a semantic segmentation framework that integrates an illumination-adaptive preprocessing pipeline with a Bayesian-enhanced U²-Net to address two core challenges in UAV-based corrosion inspection: performance degradation under uneven lighting and lack of predictive uncertainty estimation. Variational Bayesian convolutional layers are embedded in the U²-Net encoder to enhance robustness and regularization in small-sample scenarios without relying on full probabilistic inference. A high-resolution, pixel-level annotated dataset was developed for evaluation. Experimental results across 45 independent training runs show that BU²-Net achieves an F1-score of 75.088%, an IoU of 60.350%, and a recall of 71.537%, while maintaining the lowest standard deviation across all metrics. The results confirm the method's ability to improve training stability and predictive consistency under visually complex conditions. This practical integration of adaptive preprocessing and lightweight uncertainty modeling supports real-world deployment in multimodal sensing systems for intelligent structural health monitoring.

Keywords: Corrosion detection, semantic segmentation, Bayesian deep learning, U²-Net, illumination preprocessing, UAV inspection, structural health monitoring, uncertainty estimation.

Preface

This thesis is submitted to the University of Waikato, New Zealand, in fulfilment of the requirements for the master's degree in civil engineering. It has not been submitted for a degree or diploma at any other higher education institution, and to the best of my knowledge and belief, it contains no material previously published or written by another person except where proper acknowledgment is made. This research represents my own independent work. Some sentences in this thesis were refined using ChatGPT to enhance grammar and clarity; however, all content, analyses, and interpretations are entirely my own.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my primary supervisor, Lecturer Arthur Fang, for defining the overall direction of this research and providing consistent guidance throughout the project. His strategic insights and steady supervision ensured that the study progressed with clarity and purpose.

I am also deeply grateful to my co-supervisor, Professor James Lim, for his constant support and approachability. Whenever I encountered difficulties, he responded with great patience and warmth. His sincere smile and kind demeanor always brought a sense of reassurance, even during the most challenging moments. He is not only a rigorous academic but also a mentor whom I deeply respect.

Special thanks are extended to *Mr. Hanming Li*. We collaborated closely on several aspects of the project, including algorithm refinement, system integration, and deployment optimization. While all sections of this thesis were independently authored by me, our joint discussions significantly enriched the technical depth of the research.

I would also like to acknowledge *Mr. Renzhe Ma*, a member of our research group, for his assistance in collecting corrosion images during the dataset construction phase. His efforts helped lay a solid foundation for subsequent model training and validation.

Finally, I am forever grateful to my parents and my wife for their unwavering support throughout my academic journey. Your encouragement and belief in me gave me the strength to leave my professional job and return to full-time study.

This thesis would not have been possible without the support and contributions of all the individuals mentioned above. I extend my heartfelt thanks to each of you.

Table of Contents

List of Figure

List of Tables

1. Chapter 1 – Introduction

Corrosion in metallic structural elements is a common and serious defect in civil infrastructure. Forms such as stress corrosion cracking, corrosion fatigue, and temperature-induced degradation are major contributors to structural failure, accounting for approximately 42 percent of such cases [1]. Corrosion costs the global economy around 2.5 trillion USD annually, about 3.4 percent of GDP. Effective control could save up to 875 billion USD each year [2].

Traditional visual inspection of corrosion is typically performed by engineers or technicians and involves significant safety risks, including the possibility of injuries and on-site accidents. These methods are also time-consuming and costly, often relying on heavy equipment that disrupts operations and causes traffic delays [3]. In this context, automated inspection technologies, particularly those based on computer vision and machine learning, offer an effective way to improve efficiency, safety, and consistency of corrosion detection.

Accurate detection of corrosion defects has been widely acknowledged as a crucial step toward overcoming the inherent limitations of manual inspection processes and enabling more advanced maintenance strategies [4]. As civil infrastructure evolves in line with the Construction 4.0 paradigm, such intelligent visual sensing capabilities are increasingly regarded as foundational elements in supporting data-driven decision-making, predictive maintenance, and minimally disruptive operations [5]. While this study focuses on visual recognition, its function can be interpreted as contributing to the sensing layer in a larger data-driven maintenance framework.

Automatic visual inspection of corrosion and other surface damage on metal structures has become an active area of research. Corrosion, driven by continuous chemical reactions, typically appears as localized color changes and surface roughness (see Fig. 1). Corroded regions often show uneven colors and rough textures, while non-corroded areas appear smooth

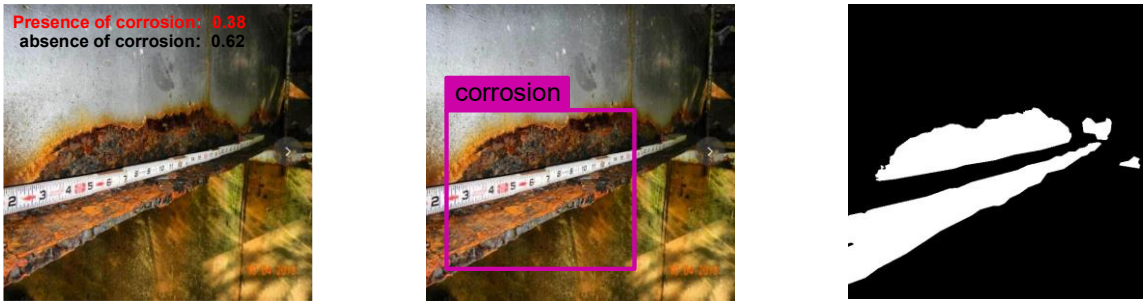
and uniform [6]. Based on these differences, rust can be detected by analyzing color distributions in different color spaces, such as HSV or HSI [7]. Wavelet-based texture descriptors have also been used to extract features for corrosion detection. These distributions can be modeled with Gaussian mixtures or grouped using unsupervised methods like k-means clustering [8]. Image analysis that combines color and texture features can help improve the detection and description of surface damage on materials [9]. These methods often fail due to their reliance on low-level features that are sensitive to noise and lighting variations. This limitation reduces their applicability in modern infrastructure scenarios that require high-precision defect detection, such as robotic maintenance, automated inspection, and prefabricated construction. Moreover, as illustrated in Fig.1, their performance significantly degrades when the corroded and non-corroded areas exhibit similar characteristics in both color and texture.

(a) Corrosion color variation (b) Original coating color (c) Background surface texture (d) Corrosion texture

Fig. 1. Visual comparison of corroded and non-corroded surfaces.

In recent years, with the advancement of artificial intelligence and deep learning, convolutional neural networks (CNNs) have been widely used to analyze RGB images for automatic visual inspection of metallic structures in structural health monitoring tasks [10]. One key advantage of CNN-based methods, compared to traditional color distribution approaches, is their higher accuracy in detecting defects. This is mainly because they are trained with annotated ground truth data. By learning from pixel-level labels, CNNs can extract more useful features and better locate the shape and size of defected areas.

In general, deep learning-based corrosion detection can be formulated as: (i) classification [10], [11], [12], [13], (ii) object detection [14], [15] or (iii) semantic segmentation [6], [16]. In the first case, the model gives a binary result with a confidence score, indicating whether at least one corroded area exists in the image (positive) or not (negative), as shown in Fig. 2a. In the second case, the model outputs bounding boxes to locate and size each corroded region (Fig. 2b). The third approach performs pixel-wise classification to generate a segmentation mask, which provides a more detailed understanding of the corrosion shape (Fig. 2c).



(a) Classification **(b) Objective detection** **(c) Semantic segmentation**

Fig. 2. Three typical deep learning tasks in corrosion recognition.

While classification and object detection are useful for identifying corrosion presence, they lack the precision needed for automated maintenance. Semantic segmentation offers pixel-level detail, including area, span, aspect ratio, and shape of corroded regions [17]. Such detailed outputs are essential for data-driven decision-making in Construction 4.0, especially in prefabrication workflows where damaged components are digitally evaluated and manufactured off-site using segmentation results [18]. Prefabrication, a core strategy in Construction 4.0, reduces on-site labor risks, traffic disruption, and overall project costs. Studies show it can shorten construction time by up to 70 percent and lower labor costs by around 43 percent [18], [19], [20].

Although semantic segmentation provides more detailed pixel-level information than classification or object detection, its application in infrastructure corrosion detection remains limited by two main challenges. One significant issue is performance degradation under uneven illumination or low-light conditions, which disrupts local feature representation and leads to inaccurate detection of corrosion boundaries [21]. Another issue is that most existing models produce deterministic outputs without accounting for pixel-level uncertainty, often resulting in high rates of false positives and false negatives that compromise the reliability of automated maintenance and decision-support systems [22].

Previous studies have explored various illumination correction techniques, including histogram equalization, Retinex-based enhancement, and CLAHE, to mitigate lighting inconsistencies in corrosion images [23], [24], [25]. Other research has investigated incorporating Bayesian inference mechanisms into CNN architectures to improve model performance by better capturing feature uncertainties [26], [27], [28]. Despite these advances, illumination normalization and Bayesian modeling have largely been pursued independently, with few attempts at developing a unified framework that addresses both challenges simultaneously. This gap defines the central scientific problem of this study: designing a corrosion segmentation approach that can adapt to varying lighting conditions, leverage Bayesian modeling for improved segmentation accuracy, and simultaneously provide pixel-wise uncertainty estimation. Such a system would enhance detection reliability and support practical deployment in real-world infrastructure maintenance scenarios.

To address this gap, we propose a two-stage, illumination-aware and uncertainty-integrated corrosion recognition framework. The first stage applies an illumination-adaptive preprocessing pipeline that combines global-local fusion (GLF) [29] and low-light image enhancement (LIME) [30] to improve consistency and clarity under complex lighting

conditions. The U²-Net architecture [31], known for its nested U-shaped design that enables effective multi-scale feature extraction for precise segmentation, is adopted as the backbone network in the second stage. In this stage, we further enhance the architecture by integrating variational Bayesian modelling into each encoder–decoder stage, enabling pixel-wise confidence estimation alongside segmentation accuracy. Unlike standard CNNs, the proposed model provides pixel-level uncertainty estimates while maintaining segmentation precision and supporting deployment on edge devices with limited resources. To the best of our knowledge, this represents the first known application of variational Bayesian modeling within U²-Net for corrosion segmentation.

Experiments on our corrosion dataset show that the proposed framework consistently improves segmentation accuracy and robustness compared to the standard U-Net and U²-Net baselines, even under varying illumination and corrosion conditions. The Bayesian-enhanced U²-Net effectively reduces false positives and false negatives by modeling prediction uncertainties, and its computational efficiency supports deployment on resource-constrained edge devices. This work represents the first integration of variational Bayesian modeling within the U²-Net architecture for corrosion segmentation. The framework simultaneously addresses accuracy and uncertainty estimation, while incorporating data augmentation techniques to ensure robust performance.

2. Chapter 2 –Related Work

2.1. Computer Vision and Machine Learning

2.1.1. Traditional Computer Vision Techniques

Corrosion usually appears in two main visual forms. First, it creates rough or uneven textures on the surface due to pitting or micro-cracking. Second, it changes the color of the surface, often producing reddish or brown tones because of oxidation. As a result, many image-based methods detect corrosion using either color features, texture features, or both. These features can be used alone or combined to improve detection performance. Other features, such as edge contours or shape descriptors, are sometimes used too, but they mostly serve as extra cues based on texture and color.

Texture was the first visual feature used in image-based corrosion detection [32], [33], [34], [35]. One basic step in texture analysis is to convert color images into grayscale. This is because most texture features are based on intensity changes, not color. Chen et al. [36] were among the first to develop image recognition methods for evaluating bridge coatings. They extracted statistical features from gray-level co-occurrence matrices (GLCM) and used a clustering method called multiresolution pattern classification (MPC) to identify surface conditions. Pidaparti et al. used wavelet transform and fractal-based analysis to classify pits and cracks in grayscale images of nickel–aluminum bronze samples [37].

When a color image is converted to grayscale, important color information is lost. Because of this, many researchers choose to extract color features directly from color images for corrosion detection. For example, Lee et al. focused on using RGB values to separate rusted areas from the background. They worked with images that had small rust spots and high-contrast surfaces [35]. The rust color was identified statistically in the red-green-blue (RGB) color space.

Ivasenko and Chervatyuk proposed a corrosion segmentation method based on the HSV color model. They applied the Retinex algorithm to correct uneven lighting and used the ICM method to label rusted regions [38]. Chen et al. compared 14 color spaces to find the best one for detecting corrosion under non-uniform lighting. They found that the a* and b* components in the Lab color space gave the best performance. [39]

Some researchers use both texture and color features to detect corrosion. For example, Gibbons et al. used a Gaussian Mixture Model trained with color features from the Lab* color space and texture features from Gabor filters [40]. Their method was designed for corrosion detection in remanufacturing processes. Khayatazad et al. developed an algorithm that uses HSV histograms and GLCM-based roughness to locate rusted regions in photographic datasets [6]. Petricca et al. developed a deep learning method using Python to automatically detect rust on metal surfaces [41]. They trained their model with more than 3,500 images.

Choi and Kim used color, texture, and shape features to classify five types of corrosion with probabilistic decision-making [42]. Medeiros et al. and Bonnin-Pascual and Ortiz also used two weak classifiers to detect corrosion in tanks, containers, and pipelines [43], [44]. The first classifier used texture roughness based on the energy property of the gray-level co-occurrence matrix (GLCM). The second classifier used color features in the HSI or HSV color space. Medeiros et al. applied discriminant analysis to define reference color ranges [45]. Bonnin-Pascual and Ortiz used trained hue–saturation (HS) histograms with different filtering methods [46].

Traditional computer vision (CV) techniques formed the earliest foundation for automated corrosion detection, relying on handcrafted visual descriptors such as color histograms, texture statistics, and edge contours. These approaches were grounded in the assumption that corrosion

manifests in distinct low-level features that can be manually isolated through thresholding, filtering, or transformation. While these methods were computationally efficient and interpretable, they exhibited limited adaptability and lacked the flexibility required for diverse real-world inspection scenarios.

As outlined in Table 1, early methods such as the GLCM-based analysis by Chen et al. (2004) demonstrated the use of grayscale texture features for coating assessment. However, this approach inherently sacrificed color information during grayscale conversion, limiting its ability to capture oxidation patterns that primarily manifest in chromatic variation. In contrast, Lee et al. (2006) employed color-based thresholding and achieved 100% accuracy in a controlled validation dataset. Yet this result came from a highly constrained environment and was built upon rigid handcrafted rules that could not generalize across lighting conditions or surface materials. Ivasenko and Chervatyuk (2010) further explored color segmentation using the HSV space and an ICM-based method, but their fixed color thresholds rendered the model fragile under non-uniform illumination and background complexity.

Fundamentally, these methods operate under deterministic rules and predefined feature spaces that assume corrosion can be reduced to static, domain-specific visual patterns. This architectural rigidity becomes a key limitation when deployed outside of controlled environments. Surface reflectivity, shadow artifacts, and overlapping materials often introduce visual ambiguity that such models cannot resolve. Moreover, because these methods decouple feature extraction from decision-making, their performance heavily depends on parameter tuning and cannot benefit from task-specific learning or feedback.

Another critical drawback is their lack of spatial reasoning. While some methods extract local texture or color distributions, they do not consider contextual relationships or structural

continuity within the image. This absence of spatial awareness prevents the model from differentiating between isolated noise and meaningful corrosion clusters, particularly in large steel surfaces or curved structures with occlusions.

To conclude, although traditional computer vision methods contributed significantly to the early stages of automated corrosion detection, their practical effectiveness is largely confined to controlled and narrowly defined scenarios. As shown in Table 1, these approaches consistently rely on handcrafted feature extraction, grayscale preprocessing, and static thresholding strategies. Such inherent constraints underscore the growing necessity for a methodological shift toward data-driven and adaptive learning frameworks, which can generalize across diverse conditions and interpret spatially complex patterns. Importantly, the shortcomings of traditional CV methods are not simply the result of outdated tools, but rather reflect deeper limitations embedded in their foundational assumptions regarding feature representation, learning mechanisms, and model generalization. Rethinking these principles is essential for building more robust and scalable corrosion detection systems.

Table 1. Summary of Traditional CV Methods for Corrosion Detection

Study	Method	Features	Accuracy	Limitation
Chen et al. (2004)	GLCM	Texture	Qualitative only	Relies on grayscale conversion; lacks adaptability
Lee et al. (2006)	Color-based thresholding	Color	100% (validation)	Small-scale dataset; relies on handcrafted features
Ivasenko & Chervatyuk (2010)	ICM-based segmentation	Color	Qualitative only	Fixed color thresholds; limited adaptability
Gibbons et al. (2014)	Gaussian Mixture Model	Color, Texture	Qualitative only	Handcrafted features; lacks self-adaptive capability

Study	Method	Features	Accuracy	Limitation
Shen et al. (2013)	Fourier filtering	Color, Fourier	Qualitative only	Fourier-based assumptions limit adaptability
Quintana et al. (2003)	EM-based color segmentation	Color	Qualitative only	Depends on Gaussian mixture assumptions; lacks flexibility

2.1.2. Classical Machine Learning Methods

To address the above limitations, researchers have adopted classical machine learning models that learn feature patterns from labeled data. SVMs have shown strong performance in rust classification. For instance, Chen et al. used SVMs to classify corroded regions in color images of steel bridges [47]. Tsutsumi et al. designed an SVM model with a radial basis function (RBF) kernel and HSV histograms to estimate corrosion levels on transmission towers, using 1,427 images for training [48]. To mitigate the effects of uneven lighting, Liao and Lee proposed a hybrid framework using least-squares SVM (LS-SVM) [8]. Additionally, Shen et al. applied Fourier transforms to image color channels before using SVM to separate rusted areas from the background [49].

K-means clustering is another widely adopted technique, especially for unsupervised segmentation of corrosion patterns. Valeti and Pakzad combined K-means in the L*a*b* space with hue-based features from HSV to enhance accuracy [50]. Quintana et al. initialized a Gaussian expectation-maximization (EM) algorithm using K-means for rust segmentation [51]. Shen et al. applied K-means clustering and used the RMSSTD metric to determine the optimal number of clusters when estimating rust coverage [52]. Other researchers explored shallow learning architectures such as decision trees or hybrid multi-stage methods. Son et al. developed a three-step approach consisting of color space conversion, decision tree–based rust

classification, and blasting zone identification [53]. Their method achieved an average classification accuracy of 97.95% across 119 test samples.

While classical machine learning models such as SVMs, K-means clustering, and decision trees have played an important role in the early stages of corrosion detection research, their effectiveness is fundamentally constrained by structural assumptions and architectural limitations. These models heavily rely on handcrafted features such as color histograms and texture descriptors, which are manually designed based on visual heuristics and lack adaptability. This reliance on static feature representations makes the models highly sensitive to variations in lighting, background clutter, and corrosion morphology, limiting their robustness in real-world scenarios.

As shown in Table 2, representative studies reflect these structural weaknesses. For example, the SVM-based approach proposed by Chen et al. (2012) achieved near-perfect results on synthetic data (F-measure = 0.999), but its performance dropped significantly on real-world images (F-measure = 0.820), highlighting its dependency on manually crafted features and its poor cross-domain generalization. Similarly, the model by Tsutsumi et al. (2009), which utilized HSV histograms to classify corrosion on power towers, achieved 85.6% average precision but remained highly sensitive to lighting changes and required manual cropping of regions of interest, making it less practical in automated applications.

Unsupervised approaches such as the K-means method proposed by Chen et al. (2018) presented additional limitations. While it enabled initial segmentation of corroded areas, its results were qualitative only and heavily affected by cluster initialization and assumptions about data distribution. It also lacked any quantitative mechanism for measuring corrosion extent. Likewise, the decision tree–based method developed by Son et al. (2014) achieved an

accuracy of approximately 97.95% on a small test set but relied on manually defined rules, making it difficult to generalize to new background conditions or deployment environments.

From a broader perspective, these models follow a segmented processing pipeline in which feature extraction, dimensionality reduction, and classification are treated as separate steps. This separation often leads to cumulative error and instability. In contrast, modern deep learning frameworks unify these stages through end-to-end optimization, enabling more coherent and resilient learning. Additionally, classical machine learning models do not incorporate uncertainty estimation, which limits their suitability for safety-critical tasks such as infrastructure inspection.

Therefore, although classical machine learning methods provide foundational baselines and remain valuable in constrained scenarios, their limited expressiveness, lack of adaptability, and structural rigidity reduce their effectiveness in high-variability or real-time corrosion detection systems. The evidence presented in Table 2 clearly illustrates a recurring pattern of handcrafted dependence, weak generalization, and shallow inference capacity, reinforcing the need for more robust and adaptive learning frameworks in this domain.

Table 2. Summary of ML-Based Corrosion Detection Methods

Study	Method	Features	Dataset	Performance	Limitations
Chen et al. (2012)	SVM	RGB, HSI	Synthetic + real images	F-measure: 0.999 (synthetic), 0.820 (real)	Needs manual feature engineering
Tsutsumi et al. (2009)	SVM	HSV histogram	1,427 images of power towers	85.6% average precision	Sensitive to lighting and manual cropping

Study	Method	Features	Dataset	Performance	Limitations
Liao & Lee (2016)	LS-SVM	HSV histogram	100 images	86% correct	Needs manual feature engineering
Chen et al. (2018)	K-means	RGB, RMSSTD	35 images	Qualitative only	No corrosion area quantification
Son et al. (2014)	Decision tree	HSI	119 test samples	Accuracy: ~ 97.95%	Sensitive to background; manual rules

2.2. Network-Based Methods

2.2.1. Shallow Networks

Unlike traditional machine learning models, Artificial Neural Networks (ANNs) are capable of learning complex nonlinear relationships between input features and corrosion-related outputs, even with limited physical modelling [54]. Therefore, ANNs are more resilient in cases where visual features between damaged and undamaged surfaces are hard to differentiate. ANNs have demonstrated strong adaptability in corrosion detection. As early as 1995, Furuta et al. used an ANN to determine threshold values for rust segmentation in the HSV color space [55]. In the same year, Livens et al. applied a Learning Vector Quantization (LVQ) network to classify corroded regions using only texture features [56].

Later, Jahanshahi and Masri proposed a corrosion detection method that combined wavelet-based texture analysis with color information, using an ANN for classification [57]. Their results emphasized that both color and texture features are essential for improving detection accuracy. Naik et al. further trained a multilayer perceptron (MLP) based solely on color

features and evaluated four different color spaces. They concluded that RGB performed the best. However, they also noted that omitting texture information limited the model's generalization [58]. In a follow-up study, Ortiz et al. optimized a single-hidden-layer ANN that integrated both color and texture features for detecting coating degradation and corrosion [3]. Shen et al. [55] developed a perception-inspired ANN model trained on intensity features, which achieved robust performance under uneven lighting and outperformed SVMs in visually ambiguous scenarios. Chen et al. used GLCM-based texture features with a shallow ANN to classify bridge coating conditions, demonstrating improved discrimination of corrosion severity levels [52].

ANNs serve as a conceptual bridge between traditional machine learning models and modern deep learning architectures in corrosion detection. These models demonstrated early promise in modeling nonlinear feature interactions, particularly in cases where rule-based methods failed to generalize. However, despite measurable improvements over handcrafted approaches, shallow networks inherit many of the same foundational limitations, primarily due to their architectural simplicity and dependence on manual feature design.

As shown in Table 3, early models such as those by Furuta et al. (1995) and Livens et al. (1995) relied on basic feature inputs like HSV thresholds and texture matrices. While their reported accuracy reached approximately 95% and 85% respectively, the performance was achieved under constrained and homogeneous settings. The absence of spatial modeling capabilities, combined with the use of fixed thresholds, rendered these systems inherently non-adaptive and difficult to transfer to variable or noisy environments. This inability to localize or reason over spatial context fundamentally limits their usefulness in practical inspection tasks, particularly those involving pixel-level defect detection.

Later efforts by Jahanshahi et al. (2013) and Naik et al. (2020) attempted to mitigate these weaknesses by introducing multi-channel color features and wavelet-based transformations. Although classification metrics improved significantly (AUC up to 0.94, accuracy ~91%), these networks continued to treat features as independent vectors, failing to incorporate spatial continuity or structural relationships inherent in image data. Even large datasets, such as the one used by Ortiz et al. (2016) with nearly 17,000 samples, did not resolve this limitation. The models still operated on fixed pipelines, without the ability to learn hierarchical or multiscale representations. As a result, performance remained highly sensitive to input design, and robustness across diverse real-world scenes could not be guaranteed.

From a theoretical standpoint, the architectural shallowness of these models prevents the encoding of complex visual patterns that span across spatial or semantic scales. Unlike convolutional architectures that extract localized features in a hierarchical manner, shallow networks rely on fully connected layers that disregard spatial topology altogether. This fundamental design constraint explains why shallow ANNs, despite improved numerical scores, often fail to generalize in practical deployment where corrosion may appear diffusely, vary in texture and scale, or be partially occluded by environmental noise.

The evidence in Table 3 illustrates a consistent bottleneck across studies: the reliance on static feature encodings, the absence of spatial reasoning, and the lack of end-to-end adaptability. These limitations suggest that while shallow networks paved the way for data-driven modeling in corrosion detection, their relevance has diminished in the context of modern requirements such as real-time segmentation, embedded deployment, and uncertainty-aware decision making. Their historical value lies not in their direct applicability today, but in laying the conceptual groundwork for deeper and more flexible architectures.

Table 3. Summary of Shallow Networks Methods for Corrosion Detection

Study	Method	Features	Dataset	Performance	Limitations
Furuta et al. (1995)	ANN thresholding	HSV	Not specified	Accuracy：~ 95%	Manual threshold; lacks adaptability
Livens et al. (1995)	LVQ network	Texture	398 images	Accuracy：~ 85%	Lacks spatial awareness
Jahanshahi et al. (2013)	ANN	Wavelet, color	2,059 images	AUC up to 0.94, Acc. 90.5%	Lacks spatial awareness
Naik et al. (2020)	MLP	RGB, HSV, CIE L*a*b	5000 train, 2064 val	Accuracy：~ 91%	Color-only input; lacks spatial features
Ortiz et al. (2016)	3-layer FFNN	Color, texture	~17,000 images	TPR 0.88 / 0.85	Lacks spatial context
Shen et al. (2018)	ANN	Color, Texture	35 images	Accuracy：~ 80%	Fixed feature extraction

2.2.2. Deep Networks

As previously mentioned, traditional machine learning has achieved considerable progress in corrosion detection. However, deep learning has rapidly emerged as a dominant approach, fueled by the increasing availability of large-scale data and significant advances in high-performance computing technologies [59]. Deep learning originates from traditional neural networks but leverages multi-layered architectures to model complex and abstract feature representations. It has demonstrated outstanding performance across a wide range of tasks, including visual recognition, speech processing, and natural language understanding [60].

Unlike traditional machine learning models that rely heavily on handcrafted features, deep learning enables automatic extraction of hierarchical representations, ranging from low-level edges and textures to high-level semantic patterns, directly from raw data [61]. This reduces the dependence on domain expertise for feature engineering and significantly improves the model's generalization capability.

Among deep learning architectures, convolutional neural networks (CNNs) are particularly prominent. Due to their ability to learn spatially invariant and discriminative features without manual intervention, CNNs have been widely adopted in image recognition and object detection tasks [62], [63]. Atha et al. used a CNN with a sliding window to classify corroded and non-corroded areas on steel surfaces. They examined the effects of CNN architecture, window size, and the use of different color spaces and grayscale images [64]. Similarly, Ma et al. applied a fine-tuned CNN combined with a sliding window technique to locate corrosion defects on ship structures [65]. Du et al. developed an improved CNN model using a parallel architecture to classify corrosion severity in grounding grids [66]. Feng et al. proposed a CNN-based approach that used magnetic flux leakage (MFL) images rather than manually measured features to distinguish damaged from undamaged regions. Their results showed better performance than traditional image processing techniques [67].

Although sliding windows are commonly used in CNN-based methods for damage localization, selecting an optimal window size is difficult because defect sizes vary across test images [68], [69]. To detect and classify multiple objects, Girshick et al. introduced Region-based CNN (R-CNN). This method uses a pretrained CNN to extract features from proposed regions and then applies linear regression and SVMs for object localization and classification [70]. Although R-CNN gives better accuracy than basic CNNs, it runs slowly because it needs to train the CNN, the regressor, and the SVM in three separate steps. To solve this problem, Fast R-CNN [71]

and Faster R-CNN [72] were developed. These models speed up training and testing by applying CNNs to the whole image and using a region proposal network (RPN) for faster region selection and end-to-end learning. However, although such object detection frameworks can localize damage, they are often not accurate enough for post hoc measurement of corrosion features.

With the advancement of deep learning, computing power, and annotated databases, image segmentation has achieved significant improvements in accuracy [73]. The Fully Convolutional Network (FCN) [74] marked a major step by removing the fully connected layers from traditional CNNs and preserving spatial information. Since then, many FCN-based end-to-end segmentation models have been developed, such as ParseNet [75], Pyramid Attention Network (PAN) [76], Feature Pyramid Network (FPN) [77], Pyramid Scene Parsing Network (PSPNet) [78], and DeepLab [79].

Some studies have applied these models for pixel-wise corrosion detection [80], [81], [82]. However, the performance of these deep networks often depends on the amount and quality of labeled corrosion data. To explore whether large but weakly labeled datasets can outperform small but expertly segmented ones, Nash et al. trained an FCN model using two types of datasets: 250 images labeled by undergraduate students and 10 images annotated by domain experts. Their results showed that training with the large weakly labeled dataset performed better [80].

Still, well-labeled public datasets for corrosion segmentation, especially for damage-level detection, are not yet available. At the same time, it remains costly and time-consuming to collect a large number of finely annotated corrosion images. Therefore, there is a need for deep learning methods that are lightweight, require fewer annotations, and still achieve accurate multi-class pixel-level detection.

One well-known model that addresses these needs is the U-Net architecture. Among deep learning models, Ronneberger et al. introduced an enhanced FCN architecture called U-Net for biomedical image segmentation, achieving strong performance even with limited training data [83]. U-Net has also been applied to semantic segmentation of corrosion damage. Katsamenis et al. trained a U-Net model using a small dataset of 116 images to detect corrosion on metallic structures [17]. Nguyen et al. proposed a U-Net-based solution for automated corrosion detection using micro aerial vehicles. Their model was trained on fewer than 40 images and achieved inference speeds of 12 fps on a single GPU [84]. Shi et al. investigated how two dataset construction methods, squeezing and cropping, affect the performance of a VGG-U-Net architecture in steel bridge corrosion segmentation [85].

The advent of deep learning has significantly transformed the landscape of corrosion detection by introducing powerful models capable of learning multi-level spatial and semantic features directly from raw image data. CNNs have demonstrated a marked shift away from handcrafted pipelines toward end-to-end trainable architectures. These models excel in capturing local textures, color gradients, and object boundaries while maintaining high adaptability across diverse visual contexts. However, despite their impressive empirical performance, deep networks also exhibit critical limitations that restrict their practical deployment and generalization in real-world corrosion inspection.

As detailed in Table 4, early CNN-based methods such as the sliding window approach used by Atha et al. (2017) achieved accuracy rates as high as 98.5% by analyzing steel surface images. However, the reliance on fixed-size local patches introduced a strong sensitivity to object scale and location. This window-based strategy lacked spatial awareness and global context, making it insufficient for tasks requiring holistic scene understanding. Similarly, Du et al. (2018) combined CNNs with Softmax and SVM classifiers for multiclass corrosion

detection and reported up to 89% accuracy. Yet their method remained dependent on preprocessed local image patches, which undermines robustness when corrosion appears across variable scales or fragmented distributions.

Subsequent architectures such as Feng et al. (2017), which utilized CNNs on magnetic flux leakage (MFL) images, demonstrated strong performance (accuracy ~98.3%) but highlighted another major drawback. These methods required large, well-labeled datasets to perform optimally, a constraint that is rarely feasible in infrastructure monitoring where annotation is labor-intensive and expert-limited. Similarly, Nash et al. (2018) employed a FCN to compare weak versus expert annotation and reported a maximum mIoU of only 0.39, with notable overfitting on the expert-labeled subset. This illustrates a broader issue in deep learning for corrosion detection: performance is highly sensitive to the quality, scale, and consistency of training data.

Attempts to resolve these challenges via advanced architectures have achieved mixed success. Nguyen et al. (2018) and Katsamenis et al. (2020) leveraged U-Net and Mask R-CNN models for semantic segmentation of corrosion regions. While accuracy reached up to 98%, the models were constrained by small training sets, resulting in limited boundary precision and unstable generalization. Shi et al. (2021) further explored dataset construction strategies using a VGG-U-Net framework and reported a moderate mIoU of ~57%. However, their performance was restricted by low damage pixel ratio and class imbalance, both of which are common in real-world corrosion datasets.

At the architectural level, CNNs are deterministic models that excel at learning hierarchical spatial features, yet they fundamentally lack mechanisms for uncertainty estimation or out-of-distribution adaptation. Additionally, conventional convolutional operations struggle to model long-range dependencies due to their limited receptive field, particularly when damage spans

large or disconnected regions. These limitations are compounded in corrosion detection, where input variation can arise from lighting artifacts, surface noise, and environmental interference.

Deep learning has substantially advanced the state of corrosion detection, particularly in automation and feature abstraction. However, the evidence in Table 4 exposes three persistent bottlenecks: (1) data dependency and overfitting in limited-sample settings, (2) weak boundary precision in segmentation tasks, and (3) insufficient uncertainty modeling for decision-critical applications. These issues indicate that future research must move beyond standard CNNs by incorporating lightweight architectures, data-efficient learning strategies, and probabilistic frameworks to support robust deployment in safety-critical field environments.

Table 4. Summary of Deep Networks Methods for Corrosion Detection

Study	Model	Dataset	Performance	Limitations
Atha et al. (2017)	CNN (sliding window)	Steel surface images (~33,000 patches)	Accuracy: Up to 98.5%	Lacks spatial context awareness, window size sensitivity
Kang et al., (2020)	SDDNet	160 images	Accuracy: 84.6%	Sensitive to variations in damage shape
Du et al. (2018)	CNN (Softmax & SVM classifiers)	10,000 preprocessed images (4 classes)	Accuracy: Up to 89%	Lacks global context; depends on local patches
Feng et al. (2017)	CNN+MFL	28,500 MFL images	Accuracy: ~ 98.3%	CNN requires large, well-labeled dataset

Study	Model	Dataset	Performance	Limitations
Nash et al. (2018)	FCN	250 weak/ 10 expert	mIoU 0.39/ 0.29	Sensitive to noise, Overfitting on small dataset
Katsamenis et al. (2020)	FCN, U-Net, Mask R-CNN	116 images	Accuracy: Up to 98%	Limited boundary precision; dataset size
Nguyen et al. (2018)	U-Net	70 images	DSC ~52.1%, Sensitivity ~75.2%	Limited by small dataset & model generalization i
Shi et al. (2021)	VGG-U-Net	200 corrosion images, 500 crack images	mIoU ~57%	Limited by small damage pixel ratio

2.3. Bayesian Methods

As mentioned earlier, Previous studies showed that deep learning models can generate semantic segmentation maps, where each pixel is labeled as either corrosion or background. So far, no publicly available labeled datasets for corrosion segmentation have been released. In addition, current models often misclassify rare or unseen objects, especially when trained on relatively small datasets. To ensure that deep learning segmentation models can be used effectively in real-world industrial applications, it is important to measure the uncertainty of their predictions. This helps to avoid unnecessary or costly decisions. Deep learning models have set new benchmarks in computer vision competitions, showing impressive accuracy gains every year. However, despite their high efficiency and strong performance, these models do not estimate

prediction uncertainty. As a result, they can fail when dealing with input images that differ from the training data, also known as out-of-distribution (OoD) data [86], [87].

Bayesian Neural Networks (BNNs) were first introduced in the 1990s [21,22] and have recently been extended to deep neural networks to support uncertainty-aware predictions in complex tasks [86], [88], [89], [90], [91]. Unlike standard deep learning models that assign fixed values to each weight, BNNs replace weights with probability distributions. This change allows the network to produce not just a prediction, but also a measure of uncertainty. In practice, the true posterior distribution of these weights is hard to compute, so approximate methods are used. The model starts with random initialization and gradually updates the weight distributions using gradient descent, forming what is known as a Bayesian approximation [31].

There are three commonly used strategies to implement Bayesian deep learning. First, variational inference replaces some weights with Gaussian distributions and samples them during each forward pass. This method allows the network to generate probabilistic outputs without major architectural changes, as demonstrated by Shridhar et al. [92]. Second, Monte Carlo dropout applies dropout during both training and inference. By randomly deactivating parts of the network and repeating the inference step, it estimates prediction uncertainty. This approach has been used to adapt models like DenseNet for semantic segmentation tasks [86]. Third, ensemble learning trains multiple models from different initial weights. These models make predictions independently, and the variation between them reflects the uncertainty. Recent studies have shown that ensembles can approximate Bayesian marginalization [93].

Bayesian deep learning models can estimate two main types of uncertainty: epistemic and aleatoric. Epistemic uncertainty comes from limited knowledge or insufficient data and can be reduced by training on more diverse samples [94], [95]. Aleatoric uncertainty, on the other hand, is caused by noise in the data and cannot be reduced by simply adding more training

examples. Kendall and Gal [86] discussed both types in detail, and many recent studies have used this classification when designing uncertainty-aware systems [90], [96], [97].

Bayesian deep learning methods introduce probabilistic reasoning into corrosion detection, offering a significant conceptual advance over deterministic models. However, their practical application continues to face fundamental challenges across algorithmic accuracy, system efficiency, and deployment feasibility.

Approximate inference techniques such as variational Bayes and Monte Carlo dropout rely on assumptions that rarely hold in high-dimensional vision tasks. These approximations often yield poorly calibrated uncertainty estimates, especially under domain shifts or image noise, limiting their trustworthiness in safety-critical scenarios like infrastructure monitoring.

High computational overhead caused by repeated sampling or ensemble inference makes these methods unsuitable for embedded systems, UAV-based inspections, or real-time monitoring applications. Existing models remain constrained to offline analyses or controlled laboratory settings, falling short of real-world performance demands.

Most Bayesian implementations inherit conventional CNN backbones that lack architectural sensitivity to corrosion-specific features such as scale inconsistency, spatial irregularity, or multi-textural diffusion. Adding uncertainty estimation does not inherently improve the model's ability to localize or resolve complex boundaries, which remains a core requirement for precise segmentation.

The theoretical distinction between epistemic and aleatoric uncertainty is rarely transformed into actionable system outputs. Many uncertainty maps remain difficult to interpret or integrate into human-supervised inspection workflows. Without validated interpretability or operational thresholds, such outputs risk undermining trust in AI-aided systems.

Despite alignment with the goals of Construction 4.0, Bayesian models are rarely integrated into complete monitoring pipelines. A lack of lightweight inference designs and unified decision-support frameworks has slowed their transition from research prototypes to deployable engineering tools.

Addressing these limitations requires more than probabilistic modeling. Future advances must couple uncertainty reasoning with architecture-level innovations, efficient inference mechanisms, and interpretable outputs tailored for decision-making under uncertainty. Only then can Bayesian methods fulfill their promise as robust and reliable components in the next generation of corrosion detection systems.

2.4. Illumination Preprocessing

The effectiveness of all previously mentioned methods is closely tied to the quality of the input images. In Bayesian neural networks, stochastic uncertainty is often associated with the inherent noise in the input signals. To reduce this type of uncertainty, the input must be modified in some way, such as by improving resolution, illuminating target regions, or capturing data from multiple viewpoints. While resolution and multi-angle acquisition can be addressed during the image capture phase, enhancing illumination can be handled through preprocessing techniques [22].

Applying a preprocessing step to correct lighting non-uniformity is often considered an ideal approach [7]. For this purpose, Retinex methods can be used together with other image enhancement techniques, because they are closely related to the psychophysical characteristics of human visual perception. Among various contrast enhancement techniques, CLAHE (Contrast Limited Adaptive Histogram Equalization) is particularly suitable for corrosion-related preprocessing due to its ability to suppress over-amplification in homogeneous regions

while enhancing local contrast. This makes it a more robust choice compared to standard AHE or global histogram equalization, especially when combined with Retinex, under conditions with severe brightness imbalance and shadow interference.

The Retinex theory, first introduced by E. Land and J. McCann, was originally developed as a model of human color perception in complex real-world scenes. It describes how each spectral subsystem in the retina contributes to vision [98]. Retinex theory is also considered a foundation for understanding color constancy [99]. In recent years, the term "Retinex" has also been used to describe methods for contrast enhancement in high dynamic range images. There are now several different interpretations of Retinex theory [100]. These can be grouped into four main types [101]: threshold-based methods developed by Horn [102], reset-based methods introduced by Frankle and McCann [103], center-surround methods [104], [105], and variational approaches [106]. Each type has its own strengths, limitations, and application areas. The center-surround Retinex comes from Land's idea that brightness can be represented as the ratio between local intensity and the average intensity in the surrounding area [107]. This approach is relatively simple and does not require thresholding or resetting, which led to the development of single-scale and multi-scale center-surround Retinex methods [104], [105].

CLAHE is an improved version of Adaptive Histogram Equalization (AHE). It enhances local contrast by processing small regions of the image independently, making local structures more visible. CLAHE is commonly used to improve image quality in applications where brightness and contrast vary significantly [108]. Originally popular in medical image processing, it is also effective for corrosion images due to their similar characteristics, such as uneven lighting and low contrast. By applying a transformation function at each pixel location, CLAHE helps reveal important details while reducing the risk of noise amplification often caused by global histogram equalization [109].

Retinex-based algorithms and CLAHE offer practical benefits for enhancing image visibility under non-uniform lighting, yet they exhibit several structural limitations that undermine their reliability in real-world scenarios. Their heavy reliance on manually tuned parameters and empirical heuristics often leads to brittle performance when deployed across diverse environments. For instance, CLAHE may inadvertently amplify noise in texture-sparse regions when local contrast thresholds are suboptimally configured, while Retinex techniques are prone to generating halo artifacts or overcompensated brightness in scenes with complex surface reflectance.

A deeper issue lies in the lack of integration between these preprocessing modules and downstream learning architectures. Most techniques are applied as fixed preprocessing routines with no adaptive feedback from model performance. This disjointed structure prevents the system from adjusting to dynamic lighting variations or model uncertainty during deployment, thereby limiting robustness in UAV-based inspections or embedded environments where lighting and corrosion morphology are unpredictable.

Static preprocessing must be reimagined as a learnable, feedback-driven component of the detection pipeline. Techniques such as joint optimization, attention-guided enhancement, or meta-learned parameters could allow illumination normalization to evolve alongside model training, increasing generalizability and segmentation stability. Only through such integration can preprocessing modules contribute meaningfully to the resilience and reliability demanded by modern infrastructure monitoring systems.

2.5. Research Gaps and Motivation

Recent advances in corrosion detection spanning traditional computer vision, machine learning, deep neural networks, illumination preprocessing, and Bayesian modeling have achieved notable milestones in laboratory conditions. However, these methods continue to face

persistent challenges when deployed in the complex and variable environments of real-world infrastructure. Classical algorithms often fail to generalize under outdoor conditions marked by irregular lighting and cluttered backgrounds. While deep neural networks have improved adaptability through hierarchical feature learning, they remain highly dependent on large, high-quality annotated datasets. Such datasets are costly to produce and often lack the diversity required to ensure robust performance under distribution shifts or unforeseen scenarios. Furthermore, most current architectures offer deterministic outputs without quantifying predictive uncertainty, limiting their trustworthiness in safety-critical applications such as bridge monitoring or structural maintenance planning.

Efforts to mitigate these limitations through illumination preprocessing, such as Retinex and CLAHE, have shown improvements in visibility and downstream detection accuracy. Yet, these techniques are typically implemented as standalone modules with little integration into the learning pipeline. This disjointed setup introduces additional variability and weakens system stability under dynamic lighting conditions. Especially under the paradigm of Construction 4.0, where autonomous corrosion monitoring is increasingly central to intelligent inspection and robotic intervention, there is a growing need for systems that go beyond accuracy, emphasizing real-time adaptability, interpretability, and deployment efficiency.

Addressing these persistent limitations calls for a unified, end-to-end approach that tightly integrates sensing, adaptive enhancement, and uncertainty-aware segmentation. In response, this study proposes a fully integrated corrosion detection framework that combines UAV-based image acquisition, illumination-adaptive preprocessing, and a Bayesian deep segmentation model. The system achieves precise pixel-level localization and predictive uncertainty estimation while maintaining computational efficiency, offering a deployable solution aligned with the demands of intelligent infrastructure monitoring and Construction 4.0.

Fig. 3 illustrates a three-stage framework. The first stage captures high-resolution images of critical steel components under complex conditions such as rusted bearings, fractured edges, and deteriorated coatings using UAVs. The second stage enhances feature visibility and consistency under uneven lighting by applying an illumination-adaptive preprocessing module that integrates GLF and LIME. The third stage performs accurate pixel-wise corrosion segmentation using BU²-Net and generates predictive uncertainty maps. The outputs support downstream engineering tasks including risk evaluation, maintenance prioritization, and prefabrication planning. The complete system supports real-time inference on edge devices and mobile robotic platforms in field scenarios.

The main contributions of this work are as follows:

1. BU²-Net demonstrates strong generalization across various lighting conditions, surface textures, and background complexity, ensuring robustness in practical inspection environments.

2. The model achieves reliable performance on a small UAV-acquired dataset, maintaining high accuracy and low variance across 45 independent training runs.

3. The GLF+LIME module improves consistency of input features under uncontrolled outdoor illumination, enhancing model stability and corrosion feature separability.

4. BU²-Net preserves inference speed comparable to conventional segmentation models while improving convergence stability during training, enabling deployment on lightweight embedded systems.

The remainder of this thesis is organized as follows. Chapter 3 describes the dataset preparation pipeline, including UAV-based image acquisition and annotation, data augmentation strategies, and protocols for dataset structuring and splitting. Chapter 4 introduces the illumination-

adaptive preprocessing module, detailing its background, the post-fusion global-local fusion approach, and an evaluation of front-end methods such as SSR, MSRCR, and LIME. Chapter 5 outlines the Bayesian segmentation framework, covering the Bayesian-U²-Net architecture with residual U-blocks and variational Bayesian integration, along with implementation considerations for training configuration and edge device deployment. This chapter also compares the framework against baseline models, specifically SVM and U-Net. Chapter 6 presents the experimental results and provides a comprehensive performance analysis across various corrosion scenarios using precision, recall, F1-score, and IoU as evaluation metrics. Chapter 7 summarizes the key findings and insights, while Chapter 8 discusses potential directions for future research.

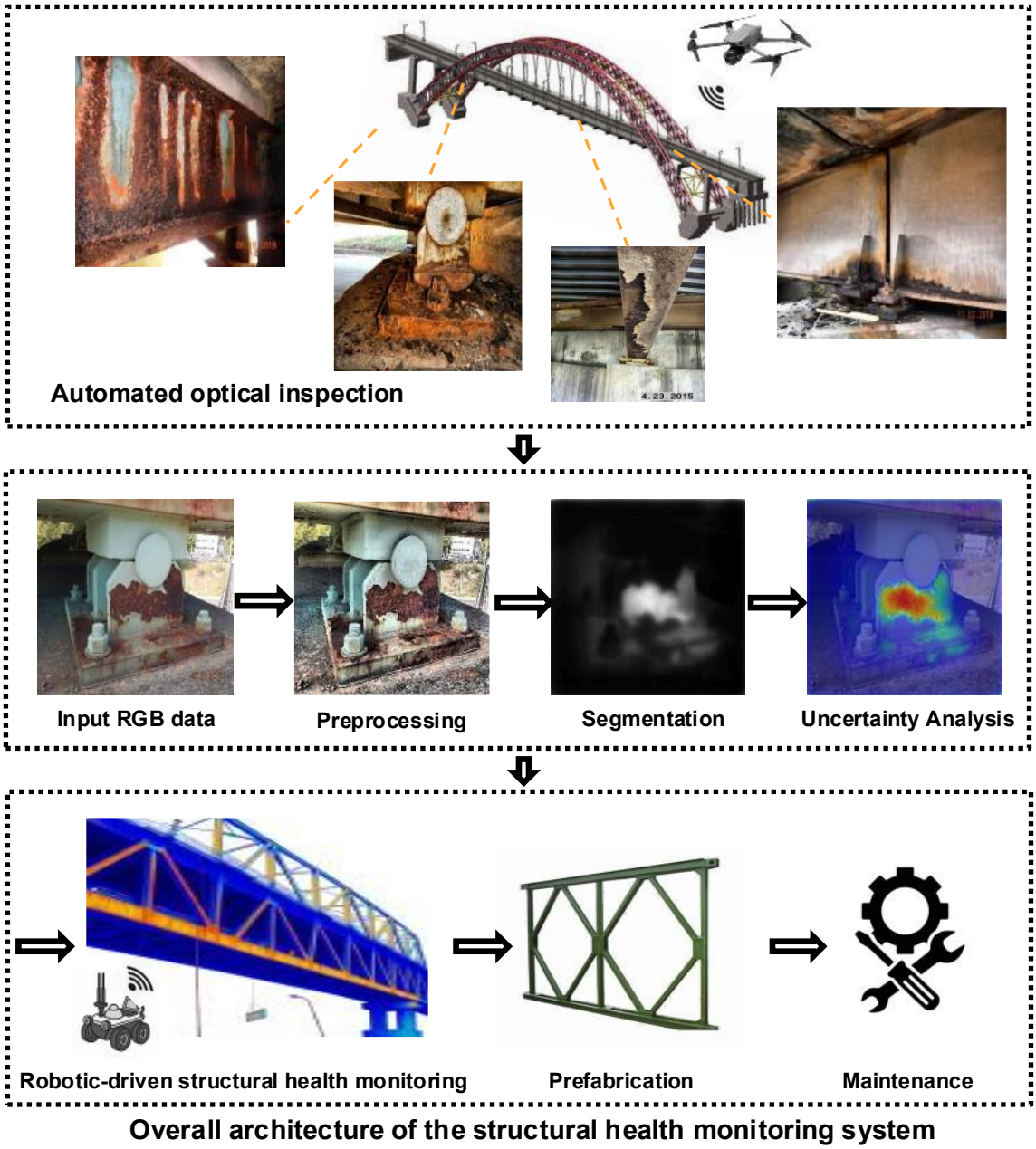


Fig. 3. Overview of the proposed corrosion detection pipeline.

3. Chapter 3 –Dataset Preparation

Chapter 3 outlines a streamlined pipeline for dataset preparation and input preprocessing, tailored to the specific challenges of outdoor corrosion recognition on steel bridge structures**.** As mentioned before, corrosion images captured in real-world environments often face challenges such as inconsistent lighting, surface contamination, and ambiguous textures. These problems can reduce segmentation performance if not properly handled.

This chapter introduces a modular data preparation framework designed to mitigate challenges related to image quality and annotation consistency. High-resolution images were acquired through UAV-based inspections, ensuring coverage of corrosion-prone areas under diverse environmental conditions. The inspection focused on corrosion-prone regions at critical joints and structural components of steel bridges, such as welded seams, bearings, and anchorage connections. In addition, this real-world dataset was supplemented with corrosion images selected from the publicly available COCO-Bridge 2021 dataset [110]. These external samples were re-annotated to match the pixel-level precision and annotation protocol used in this study.

The pipeline includes image quality screening, expert-reviewed annotation, and targeted data augmentation to improve generalization. A reproducible dataset partitioning protocol is also used to ensure consistent training and fair evaluation. These steps build a strong data foundation for the segmentation models in the following chapters.

3.1. Image Acquisition and Annotation

This study constructed a corrosion image dataset containing 300 samples, specifically curated to evaluate semantic segmentation performance under small-data conditions. Images were acquired from multiple steel bridge sites across New Zealand’s North Island using a DJI UAV equipped with an optical camera. Key structural elements, including girders, cross-bracing, welded joints, and bearing plates, were targeted to capture representative corrosion patterns

[111]. A standoff distance of approximately one meter was maintained during image acquisition to ensure operational safety and adequate resolution.

The field-collected dataset was complemented by selectively re-annotated samples from the COCO-Bridge benchmark, which enhanced the visual and structural diversity. The final dataset maintains a deliberately limited scale while capturing realistic inspection conditions. This controlled setting facilitates the evaluation of uncertainty-aware and regularization-based segmentation models under low-data constraints.

Structural complexity and environmental variation were addressed by capturing each target area from multiple viewing angles and flight altitudes. This strategy enhanced spatial coverage and allowed the system to document corrosion features under varying lighting conditions and common visual obstructions typically encountered in real-world bridge inspections. Fig. 3 presents representative target locations on critical steel bridge components used during data collection [112].

Images with motion blur, poor lighting, or occlusion were manually removed during post-processing [113]. About 500 images were first collected, and 300 images were kept after strict quality checks. All selected images were resized to 320 × 320 pixels and saved in 24-bit RGB TIFF format to keep high image quality and to fit image processing tools.

Dataset quality during training was maintained by retaining only images that exhibited clear corrosion features and favorable lighting conditions. Illumination preprocessing techniques, detailed in Section 3.2, were not applied directly to the dataset itself. Instead, these methods were integrated into the model's input pipeline to improve robustness under varying illumination scenarios during deployment [114].

Table 5. Image Quality Filtering Criteria Applied During Dataset Preparation

Criterion	Description	Action Taken
Motion Blur	Loss of edge sharpness due to drone vibration or movement	Discard
Poor Lighting	Underexposed or overexposed regions with low contrast	Discard
Occlusion	Visual obstruction from vegetation, cables, or drone parts	Discard
Resolution Mismatch	Abnormal aspect ratio or scaling artifacts	Resize
Visual Noise/Artifacts	Sensor noise, rain drops, glare, or compression artifacts	Discard
Lens Distortion	Severe warping or fisheye distortion	Discard

Note: These criteria were applied manually during post-processing to ensure consistency, clarity, and suitability of the retained samples for pixel-level annotation and segmentation model training.

Fig. 4. Sample images illustrating common quality issues such as glare, occlusion, rust artifacts, vegetation obstruction, poor lighting, and motion blur.

After image selection, the remaining samples were annotated at the pixel level using Roboflow [115], a cloud-based annotation platform. Although only high-quality images with clear corrosion features were kept, the annotation process still showed some challenges caused by lighting inconsistencies and shadow artifacts [116] These findings guided the design of the illumination normalization module introduced later in the pipeline (Section 4.2), which helps improve model robustness under real-world deployment conditions.

High annotation quality was ensured through a two-stage labeling protocol designed to improve accuracy and consistency across samples [117]. The initial annotations were completed by a structural engineer with field experience and then reviewed and refined by a second expert. On average, it took about eight minutes to annotate each image. A visual reference guide showing typical corrosion features was given to all annotators to help maintain inter-observer consistency.

This section describes the construction of a corrosion image dataset for steel bridge inspections. The dataset was developed through UAV-based image collection, quality screening, and expert-reviewed pixel-level annotation to ensure consistency and applicability [118]. The dataset combines original field images and re-annotated samples from a public benchmark dataset to balance real-world variability and annotation consistency. A selected subset of COCO-Bridge 2021 images was manually re-annotated to match the pixel-level protocol used in this study.

The structured preparation process supports the training and validation of advanced segmentation models and ensures reproducibility for future research. Together, these efforts build a solid and transferable foundation for the model design discussed in Chapter 4.

3.2. Data Augmentation

In this study, corrosion inspection images were collected under complex outdoor conditions using UAV-mounted cameras, resulting in variations in illumination, surface texture, and viewing angles. Due to the high cost and manual effort required for pixel-level annotation, only a limited number of high-quality labeled images were available. The small dataset size poses challenges for training deep segmentation models, including increased risk of overfitting and limited generalization to unseen conditions [113], [119].

To address the limitations of a small dataset without requiring additional annotations, a structured data augmentation strategy was developed, tailored to UAV-based corrosion inspection scenarios. By simulating realistic geometric and photometric variations, the training data distribution was synthetically expanded, increasing both the diversity and the effective sample size encountered by the model during training.

Variation in inspection viewpoints and structural symmetry was addressed through the application of three geometric transformations to each training sample, with selection guided by tuned probabilities. First, random flipping along the horizontal and vertical axes was applied to simulate bidirectional UAV flight paths and mirrored corrosion patterns commonly observed on opposite sides of steel components such as girders and bracing members. Second, random rotations within a ±15° range were used to model camera tilt and angular variation introduced during UAV or handheld inspections. Third, scaling transformations replicated the effect of changing distances between the camera and target surfaces, enabling the model to learn corrosion features across multiple spatial resolutions. All geometric operations were simultaneously applied to both the input image and its corresponding annotation mask to preserve spatial alignment.

Besides geometric changes, appearance-based variability was added to simulate environmental and sensor interference. Gaussian noise with zero mean and a standard deviation of 5 was used to mimic surface dust, low-light grain, or sensor distortions. Color jittering techniques were also applied, changing brightness (±12%), contrast (±15%), and hue (±5%) within set limits. These changes help stop the model from relying only on color or lighting cues and guide it to focus more on the structural and textural features of corrosion.

All augmentation operations were done using the Albumentations library in an online (on-the-fly) way [120]. During training, image–mask pairs were changed in memory as they were loaded, and each epoch created new random versions of the same base samples. This method does not change the original dataset or increase disk space, but it lets the model see more visual conditions through random changes at runtime. Online augmentation is now a common practice in deep learning-based semantic segmentation, especially when labeled data is limited. Each augmentation was applied randomly during the training phase and turned off during validation and inference, following standard deep learning practices. This setup helps the model learn to generalize across a wider range of inputs without affecting the fairness of evaluation.

As shown in Fig. 5, geometric and photometric augmentations were applied to generate multiple visually distinct versions of the same base image. These transformations simulate variations in illumination, orientation, and scale commonly observed in UAV-based corrosion inspection, effectively increasing the diversity of training samples without additional data collection.

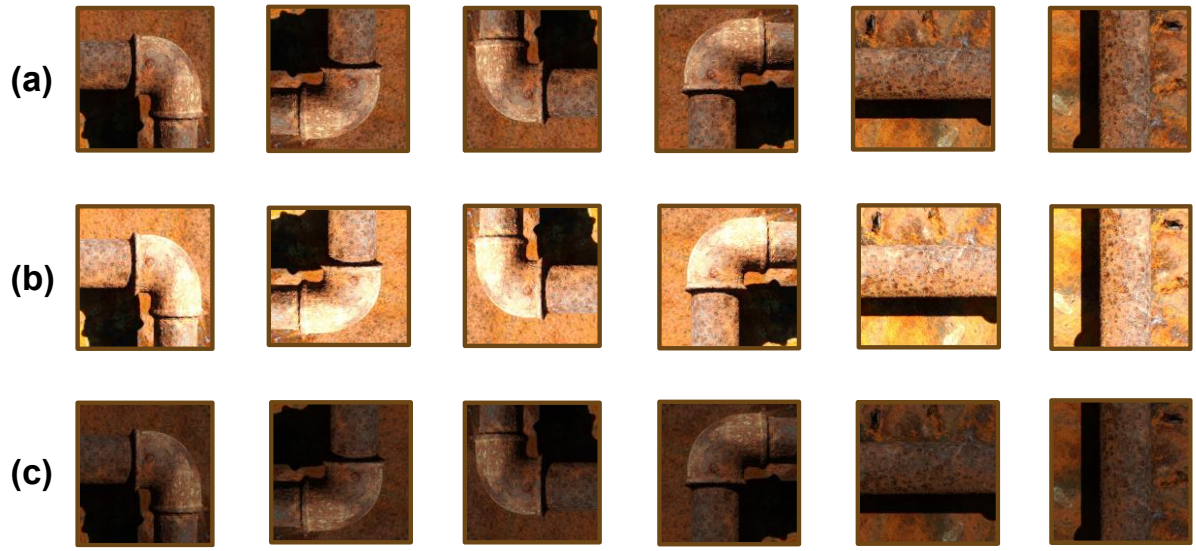


Fig. 5. Example of data augmentation for corrosion patches: (a) Original samples. (b) Geometrically transformed samples. (c) Photometrically enhanced samples to improve model generalization.

3.3. Dataset Structuring and Splitting

Consistency and reproducibility were ensured by organizing the dataset into a standardized structure. Each input image was paired with a pixel-level grayscale segmentation mask under a unified naming convention. The dataset was divided into three folders: /images/ for raw images, /masks/ for binary masks, and /splits/ for partition files (train.txt, val.txt, test.txt). This structure supports easy integration with PyTorch-based segmentation frameworks and preserves experimental integrity, especially given the small dataset size.

A total of 300 labeled images were used in this work. Stratified random sampling was applied to maintain a balanced distribution of corrosion conditions across the training, validation, and test subsets. The dataset was divided into a training set of 210 images (70%), a validation set of 45 images (15%), and a test set of 45 images (15%). A fixed random seed ensured that the data splits remained reproducible. Label distributions within each subset were manually

reviewed to verify that class imbalance was minimal. All split lists were stored as plain-text index files for convenient access during model training and evaluation.

Each segmentation mask was saved as a single-channel grayscale image. Pixel values were set as 0 for background (non-corroded areas) and 1 for corroded regions, making it a binary semantic segmentation task. The annotations were created and exported using the Roboflow platform in semantic segmentation mask format (PNG), with class indices stored directly in the pixel values. A custom post-processing script checked pixel consistency and converted the masks into the final index format needed for deep learning model training.

All input images were resized to 320 × 320 pixels at runtime to support model deployment in UAV-based real-time corrosion inspection scenarios. This resizing ensured compatibility with the input dimensions required by the segmentation model, while also reducing memory consumption and improving inference speed on edge computing platforms such as Jetson, Raspberry Pi, and TensorRT modules. File directory structures and naming conventions were aligned with the deployment system's I/O pipeline to enable seamless image loading and prediction output.

The data loading pipeline includes an intermediate stage between disk reading and input resizing, allowing the insertion of preprocessing hooks. This design supports illumination normalization techniques such as CLAHE, Homomorphic Filtering, or Retinex-based methods. These algorithms enhance image contrast and correct lighting inconsistencies before the segmentation process begins. Within the PyTorch framework, this was implemented as an optional module inside the transforms.Compose() sequence. By enabling or disabling this module, the pipeline accommodates different experimental setups or real-time deployment needs without altering the base dataset.

4. Chapter 4 –Illumination Preprocessing

4.1. Motivation and Design

Corrosion images taken in outdoor environments, especially by UAVs, often have lighting problems like partial shadows, strong reflections, and dark areas. These issues reduce edge visibility and cause style differences between images, which can hurt the stability of learning-based segmentation. To fix this, we propose a two-stage preprocessing system made of a local enhancement module and a global style normalization module. This chapter introduces the reasons for the design, the structure of the system, and the test results.

Real-world corrosion images, especially those taken under bridges or in shadowed industrial areas, often have strong non-uniform lighting because of complex environmental conditions [121],[122]. Parts of the same image may show deep shadows and bright overexposure at the same time, which breaks visual consistency and feature continuity. These problems make it harder for semantic segmentation models to learn, because the models need stable pixel-level features across the training data [123].

While neural network-based enhancement methods exist, they are often hard to use in UAV systems because they need heavy computing and their outputs are not always consistent. Also, the enhanced results can vary in style between samples, which breaks the uniformity that convolutional segmentation models need. On the other hand, classical enhancement methods are lightweight and easy to understand, but they often fail to keep structural details and cannot handle dynamic range changes across different scenes.

As part of the front-end enhancement module, three classical illumination correction algorithms were selected as candidate methods: Single-Scale Retinex (SSR), Multi-Scale Retinex with Color Restoration (MSRCR), and Low-Light Image Enhancement via

Illumination Map Estimation (LIME) [124], [125], [126]. These algorithms are widely used in low-light and uneven illumination scenarios due to their clear mechanisms and stable outputs [127]. Among them, LIME was selected as the primary enhancement method, and its effectiveness was validated through ablation experiments using SSR and MSRCR as comparison baselines.

Challenges related to uneven illumination and visual inconsistency were addressed through a task-specific, two-stage preprocessing framework designed to enhance feature clarity and input consistency. The front-end applies the LIME algorithm to enhance structural features in shadowed regions while preserving edge continuity. The back-end employs a fixed Global-Local Fusion (GLF) module to improve inter-sample consistency, suppress overexposure, and normalize image tone using a variational fusion strategy. The overall design balances enhancement effectiveness with deployment feasibility in UAV-based corrosion inspection scenarios.

4.2. Post-Fusion: GLF

A fixed post-processing step, known as GLF, was applied after all front-end enhancement methods to improve style consistency, suppress local overexposure, and preserve structural details near corrosion boundaries. GLF is a variational image fusion model first proposed by Tian and Cohen [29], designed to combine global brightness correction with local detail preservation. As a non-learning-based method, it does not need training, runs fast, and fits well with low-resource systems like UAV-based corrosion inspection.

Fig. 6 shows the visual and statistical results from each stage of the enhancement process. From left to right, the images are the original image I, its linearly stretched version $\tilde{I}$, the global

enhancement result *G*, the local enhancement result *E*, and the final fused output *F*. Each image is paired with a grayscale histogram that reflects how brightness values are distributed.

The original image *I* is underexposed and has a narrow histogram range. The stretched version $\tilde{I}$ slightly increases contrast but does not bring out much detail. The global result *G* increases brightness but reduces structural clarity. The local result *E* sharpens textures and corrosion edges, but it also introduces overexposed regions. The fused output *F* gives a balanced visual result, with an even histogram spread and clear structural boundaries.

$$F = \arg\min_{Z} \left\{ \alpha \sum_{x} \omega_G(x)(Z(x) - G(x))^2 + \omega_E(x)(Z(x) - E(x))^2 \right\}$$

Where $Z(x)$ is the fused pixel value, $G(x)$ and $E(x)$ are pixel-level adaptive weights for global and local branches respectively, $G(x)$ and $\omega_E(x)$ are their corresponding enhanced values, and α is a scalar balancing factor.

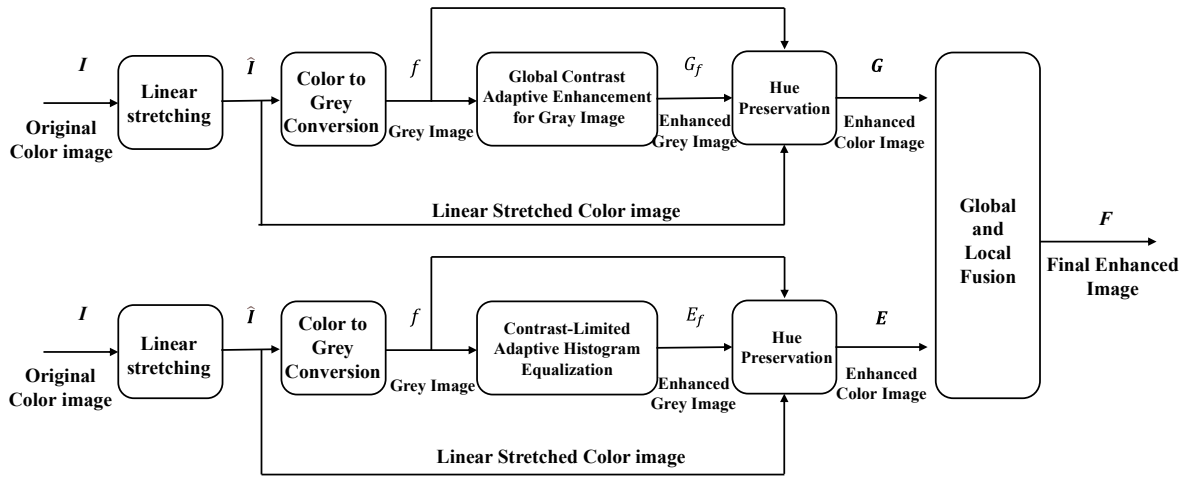


Fig. 6. Processing flow of the GLF module, including original input I, linear stretch Iˆ global enhancement G, local enhancement E, and final fusion output F.

In our implementation, the GLF module was configured with parameters adapted for corrosion images captured under uneven lighting conditions. The balancing factor α was set to 1.0, which gave equal importance to both the global and local enhancement branches. This neutral setting avoided bias toward either brightness correction or texture sharpening and resulted in consistent visual quality in both shadowed and well-lit regions.

The linear stretch gain γ was set to 1.8 based on histogram analysis of raw UAV images. This value expanded the dynamic range in underexposed areas while preventing global oversaturation, which tended to appear when γ exceeded 2.0.

For the local enhancement branch, the patch window size *k* was set to 5×5. This size helped capture edge and corrosion boundary structures effectively while reducing the amplification of background noise. The pixel-wise fusion weights $\omega_G(x)$ and $\omega_E(x)$ were computed from local luminance deviation and structural contrast. A small constant ε = 1e−6 was added during normalization to ensure numerical stability.

Evaluation Metric

We also report the grayscale standard deviation (Std) at each stage, which quantitatively characterizes image contrast. Std is particularly suitable for evaluating low-light enhancement tasks where reference images are unavailable, as it directly reflects the global tonal spread and contrast variation of the image without relying on ground-truth comparisons. It is defined as follows:

$$\text{Std} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Where x_i is the intensity of the i^{th} pixel, μ is the mean intensity, and N is the total number of pixels. A higher Std indicates stronger contrast and sharper structural variation.

As shown in Fig. 7, while global and local enhancement stages yield higher Std values, they are often accompanied by over-enhancement or loss of semantic integrity. In contrast, the fused output F achieves an ideal balance between contrast improvement and tonal stability, verifying the GLF module's effectiveness in maintaining perceptual quality and structural coherence.

GLF functions not only as a style normalizer but also as a lightweight structure-preserving fusion mechanism. By enforcing uniform brightness and preserving semantic detail, it plays a central role in our preprocessing framework and is consistently applied across all enhancement strategies in this study.

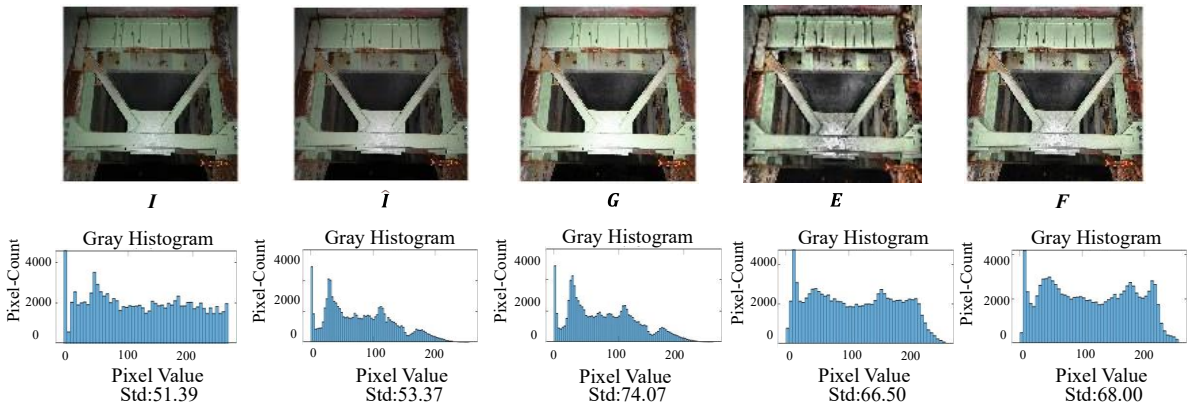


Fig. 7. The final fused result F, evaluated via intermediate outputs, histograms, and standard deviation, shows enhanced contrast and structure preservation, validating the GLF strategy.

4.3. Front-End Methods

4.3.1. SSR

Three classical non-learning-based illumination enhancement methods, Single-Scale Retinex (SSR), Multi-Scale Retinex with Color Restoration (MSRCR), and Low-Light Image

Enhancement via Illumination Map Estimation (LIME), were implemented as comparative baselines to support the selection of the front-end enhancement algorithm.

SSR is the simplest and most classical form of the Retinex theory and serves as a low-complexity illumination correction module in our preprocessing pipeline. It assumes that an image $I(x)$ consists of a reflectance component $R(x)$ and an illumination component $L(x)$, modeled as:

$$I(x) = R(x) \cdot L(x)$$

SSR approximates the illumination by applying Gaussian smoothing to the input image and computing the reflectance in logarithmic space, as defined below:

$$R(x) = \log\, I(x) - \log\,(I(x) * G_\sigma(x))$$

In this formulation, $G_\sigma(x)$ represents a Gaussian kernel with standard deviation σ = 90, and * denotes convolution. This approach suppresses low-frequency illumination components while preserving high-frequency structural details.

As shown in Fig. 8, when combined with the GLF module, SSR yields visually acceptable enhancements, especially in well-lit scenes. Nevertheless, the corresponding grayscale histogram exhibits a relatively narrow distribution, with a standard deviation of 54.15. This indicates low tonal diversity and a tendency toward grayish, low-contrast outputs. Such compression effects may obscure fine corrosion boundaries or reduce feature salience, particularly on metallic surfaces with subtle reflectance variations.

Despite these limitations, SSR remains a useful baseline due to its simplicity, low resource demands, and fast runtime. It is suitable for benchmarking other enhancement methods in resource-constrained systems or real-time pipelines where advanced correction is not critical.

4.3.2. MSRCR

The Multi-Scale Retinex with Color Restoration (MSRCR) is introduced as a contrast enhancement baseline to overcome the limitations of Single-Scale Retinex (SSR), particularly its sensitivity to filter scale and its tendency to produce grayish visual outputs. MSRCR is designed to enhance structural details while preserving the perceptual naturalness of color under complex lighting conditions.

The reflectance map is computed by averaging the log-difference between the input image and its blurred versions across multiple Gaussian kernels:

$$R_{MSR}(x) = \sum_{i=1}^{n} w_i \left[\log I(x) - \log \left(I(x) * G_{\sigma_i}(x)\right)\right]$$

Here, $I(x)$ denotes the input image, $G_{\sigma_i}(x)$ is a Gaussian kernel with standard deviation σ_i and w_i is the corresponding fusion weight. In our implementation, we use three Gaussian scales with $\sigma \in \{15,80,250\}$, and assign uniform weights $w_i=\frac{1}{3}$ to each. This multi-scale formulation enables the model to simultaneously capture fine texture and broad illumination gradients.

MSRCR applies a pixel-wise color restoration function to counteract the desaturation effect introduced by logarithmic compression, as defined below:

$$I_{MSRCR}(x) = C(x) \cdot R_{MSR}(x), \qquad where\ C(x) = \beta \cdot \log\left(\alpha \cdot I(x)\right)$$

Here, $C(x)$ restores local chromaticity using a logarithmic scaling of the input intensity. We follow the classical parameter settings with $\alpha = 125$, $\beta = 46$. All Gaussian smoothing operations are performed using MATLAB's imgaussfilt for efficient computation.

From a visual perspective (Fig. 9), MSRCR improves brightness uniformity in shaded, metallic, and reflective corrosion regions, particularly when used in conjunction with GLF post-processing. However, its grayscale histogram reveals a high standard deviation (Std = 61.32),

indicating potential tonal redundancy and signs of over-enhancement. We also observed that MSRCR tends to amplify mid-tone noise and exaggerate high-contrast edges, occasionally producing color shifts or halo artifacts.

Due to its strong sensitivity to parameter values and limited robustness under rapid lighting variations, MSRCR is not ideally suited for UAV-based real-time inspection tasks. Nevertheless, it remains a valuable benchmark for testing log-based multi-scale enhancement schemes, especially in scenarios where processing time is not a primary constraint.

4.3.3. LIME

LIME enhances low-light images by estimating a pixel-wise illumination map $T(x)$, under the Retinex-based decomposition model:

$$I(x) = R(x) \cdot T(x) \Rightarrow R(x) = \frac{I(x)}{T(x) + \epsilon}$$

Here, $I(x)$ is the observed input image, $R(x)$ is the reflectance component, and $T(x)$ represents the illumination map. A small constant $\boldsymbol{\epsilon} = 10^{-3}$ is added to prevent division by zero during reflectance recovery.

To estimate $T(x)$, LIME first constructs an initial approximation $T_0(x)$ using a channel-wise maximum operation:

$$T_0(x) = \max_{c \in \{R,G,B\}} I^c(x)$$

In this formula, $T_0(x)$ denotes the initial estimation of the illumination map at pixel x.The variable $I^c(x)$ represents the intensity of the input image $I(x)$ at pixel x and color channel c, where $c \in \{R, G, B\}$. Although it may be less accurate in scenes with strong color bias or low texture, it is computationally efficient and suitable for real-time systems such as UAV-based inspections.

LIME enhances spatial consistency while preserving edge structures by applying a structure-aware refinement to generate the final illumination map T^*, as defined below:

$$T^* = \arg\min_T \left\{ \| T - T_0 \|^2 + \lambda \sum_x \| \nabla T(x) \| \cdot \exp\left(-\gamma \| \nabla I(x) \|\right) \right\}$$

In this formulation, T^* denotes the optimized illumination map, refined from the initial estimate T_0. The first term $\| T - T_0 \|^2$ enforces fidelity to the original illumination estimate, while the second term introduces a structure-aware regularization. Specifically, $\nabla T(x)$ denotes the gradient of the illumination map at pixel x, and $\nabla I(x)$ is the gradient of the input image. The exponential weighting $\exp(-\gamma \| \nabla I(x) \|)$ suppresses smoothing near image edges, allowing the illumination map to preserve transitions along strong structural boundaries.

The parameters λ and γ are regularization weights controlling the trade-off between smoothness and edge preservation. In our implementation, we empirically set λ=0.15, γ=0.8, which produced visually consistent illumination maps across corrosion types.

We implement LIME using MATLAB, first applying imguidedfilter for fast smoothing, then using pixel-wise division and grayscale normalization. The enhanced output shows balanced lighting without changing the structure, as seen in Figure. 8 and Figure. 9. The grayscale histogram stays unimodal, with a standard deviation (Std) of 62.14, showing strong dynamic range compression and good contrast enhancement.

4.3.4. Evaluation

We compare five typical setups on a low-light corrosion image to evaluate the effectiveness of the proposed enhancement strategy. The results are shown in Figure 8, including: (a) the original input, (b) CLAHE+LIME (without GLF), (c) MSRCR+GLF, (d) LIME+GLF, and (e) SSR+GLF. Among them, CLAHE+LIME acts as the contrast-only baseline to compare with

GLF+LIME, while the other three methods all use GLF as a fixed post-processing step, differing only in the front-end enhancement method.

Each method shows different visual features. CLAHE+LIME boosts local contrast but does not control global brightness well, leading to very bright areas and lost texture details. MSRCR+GLF makes dark areas clearer but causes unnatural colors and oversaturation. SSR+GLF gives even brightness and clear structure but shows slight blurring around corrosion edges. In contrast, LIME+GLF keeps corrosion edges sharp, improves visibility across all areas, and maintains a smooth visual style, reaching a good balance between visual quality and structural accuracy.

Fig. 9 presents the grayscale histograms and corresponding standard deviation (Std) values for each enhanced image as quantitative support for the above observations. Std reflects the global tonal distribution and is particularly useful for assessing contrast enhancement in low-light scenes. Unlike SSIM and PSNR, which require reference images, Std is a reference-free metric and is therefore more suitable for our dataset, where ground-truth illumination conditions are unavailable. This makes it a practical and interpretable choice for evaluating illumination normalization in corrosion inspection tasks.

The original image (a) exhibits the highest Std (62.07), indicating strong global contrast but poor detail visibility in underexposed areas. Applying CLAHE+LIME (b) reduces the Std to 58.39, improving local contrast but introducing tonal saturation, as shown by dual histogram peaks near the high end. MSRCR+GLF (c) achieves a slightly lower Std of 61.32 with a more centered histogram, but it suffers from increased noise in midtones, which may affect boundary consistency. LIME+GLF (d) delivers the most perceptually balanced enhancement. Although its Std is relatively high (62.14), the histogram remains smooth, unimodal, and evenly

distributed, which suggests effective dynamic range compression without causing structural distortion. In contrast, SSR+GLF (e) produces the lowest Std (54.15), but the flatter histogram and slight loss of edge clarity suggest mild over-smoothing. Taken together, these findings reinforce the selection of LIME+GLF as the optimal configuration because it provides a favorable trade-off between tonal spread, contrast enhancement, and semantic boundary preservation.

These quantitative trends align well with the visual results in Fig. 8 and further justify the selection of LIME+GLF as the primary enhancement strategy adopted in this study.

(a) Original Image (b) CLAHE+LIME (c) GLF+LIME (d) GLF+MSRCR (e) GLF+SSR

Fig. 8. Visual comparison of five brightness enhancement strategies on a low-light corrosion image: (a) Original. (b) CLAHE+LIME without GLF. (c) MSRCR+GLF. (d) LIME+GLF (proposed). (e) SSR+GLF.

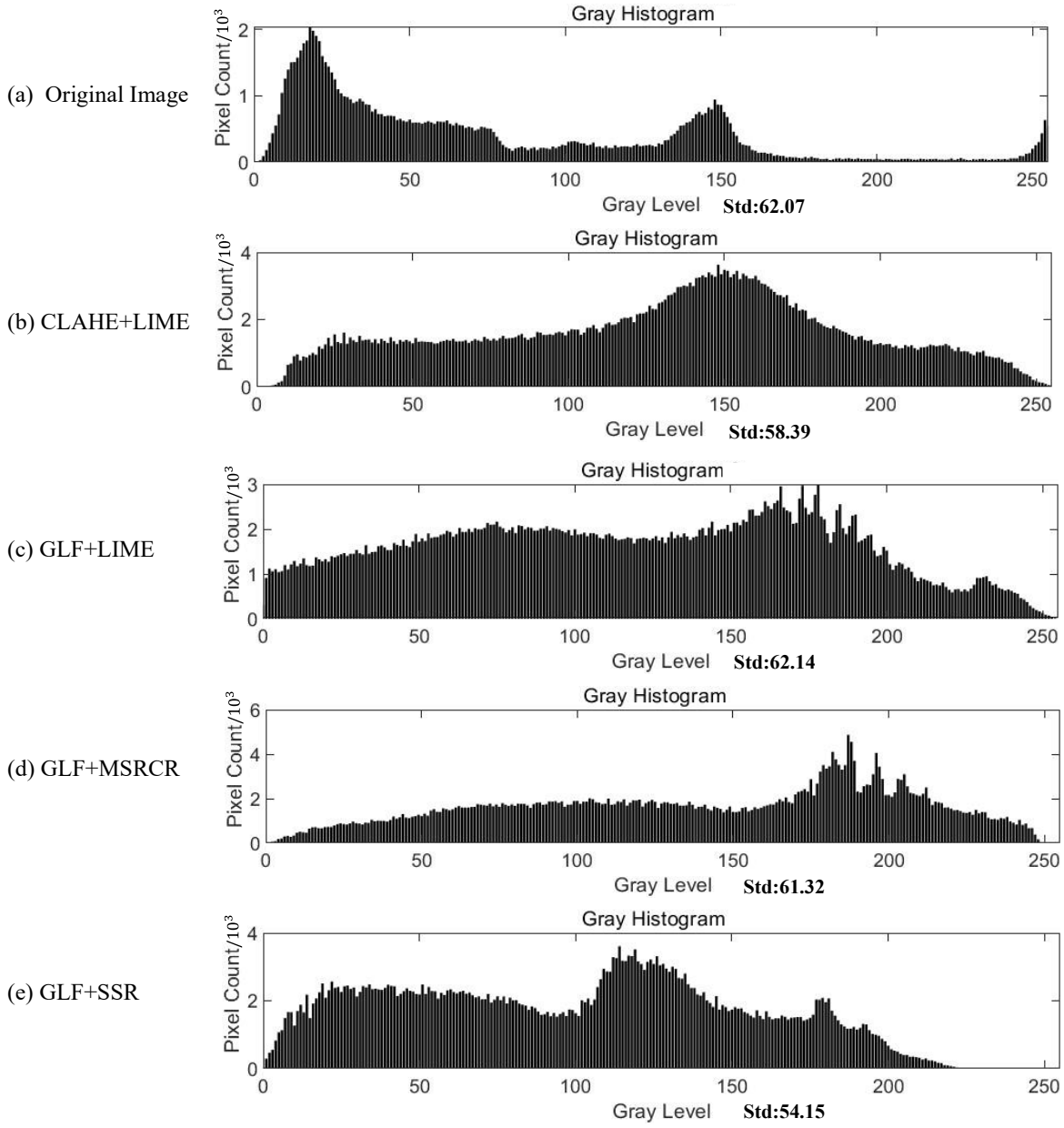


Fig. 9. Grayscale histograms and standard deviations show that LIME+GLF produces a smoother, more balanced distribution and achieves an optimal balance between contrast and structural clarity, supporting its use as the default enhancement method.

4.4. Summary

Following the comparative evaluation presented in this chapter, the GLF+LIME configuration was selected as the default preprocessing strategy because it consistently outperformed other methods across diverse corrosion scenarios. Compared with lightweight Retinex-based variants

such as GLF+SSR and GLF+MSRCR, the GLF+LIME pipeline produced enhanced images with more coherent brightness distributions, sharper edge structures, and greater stylistic consistency under varying lighting conditions.

Fig. 10 presents representative results from three challenging input cases that include severe shadows, specular reflections, and overexposed regions. The figure shows the original image, the enhanced output, and the corresponding grayscale histogram. The enhanced outputs display perceptual uniformity and well-preserved structural features, while the histograms exhibit smooth, unimodal distributions without saturation spikes or tonal collapse. These results confirm that GLF+LIME can compress dynamic range, reduce lighting artifacts, and harmonize tonal style while preserving important texture and boundary information.

The enhanced images generated by this configuration are used as standardized inputs for the Bayesian-U²-Net segmentation model, as described in the following chapter. This consistent preprocessing stage contributes to the stability and reliability of downstream pixel-level analysis, especially in real-world environments where lighting and surface reflectance vary significantly.

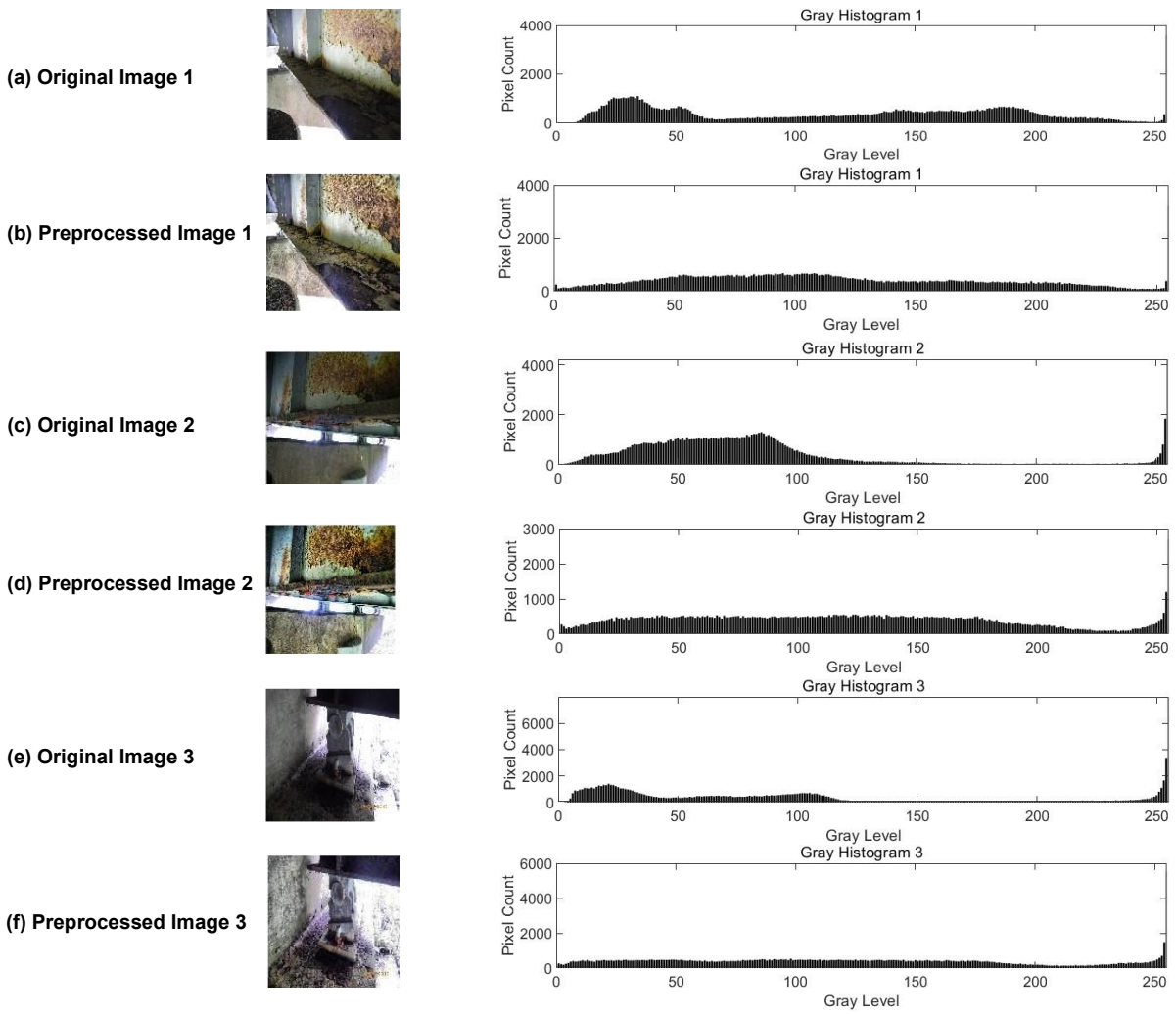


Fig. 10. Original images, enhanced results, and grayscale histograms for three corrosion cases using the GLF+LIME method.

5. Chapter 5 –Bayesian Segmentation

5.1. Motivation and Design

Building on the illumination-aware preprocessing strategies established in Chapter 4 [29], [106], which normalized lighting and contrast across the dataset, we obtain a cleaner and more consistent input distribution for segmentation. However, even with improved image quality, the task of semantic segmentation for surface corrosion remains inherently challenging due to its irregular geometries, low inter-class contrast, and visually noisy boundaries that are further compounded by occlusion and structural interference [128].

Traditional architectures like U-Net, with shallow encoding depth and coarse skip connections, often struggle to preserve edge-level details and tend to overfit in small, domain-specific datasets, where annotated training samples are limited and visually diverse [129]. This limitation is particularly pronounced in corrosion segmentation tasks, where collecting pixel-level labels is both time-consuming and resource-intensive due to the irregular, ambiguous nature of corrosion patterns and the harsh conditions of real-world inspections. As a result, available datasets are often small and highly imbalanced, which amplifies the risk of poor generalization. To address these challenges, we adopt U²-Net [30] as our architectural baseline. With its nested U-structure and Residual U-blocks (RSUs), U²-Net has demonstrated superior performance in salient object detection, offering stronger multi-scale feature extraction and improved boundary preservation under limited data conditions.

While U²-Net serves as a strong architectural baseline due to its multi-scale representation and boundary-preserving capabilities, its deterministic nature still leaves it susceptible to overfitting in small-data regimes, particularly when corrosion patterns exhibit high ambiguity and variability. To further enhance robustness under such conditions, we propose Bayesian-

U²-Net, an extended version of the original U²-Net architecture. In this design, we incorporate variational Bayesian convolutions (BBBConv2d) into the encoder path, replacing standard deterministic layers with probabilistic ones based on the Bayes by Backprop framework [31]. This allows the model to learn distributions over weights, introducing parameter variability and enabling implicit regularization through posterior sampling, thereby improving generalization and mitigating overconfidence in visually ambiguous or underrepresented regions.

5.2. Bayesian-U²-Net

Bayesian-U²-Net is an enhanced segmentation model based on the original U²-Net architecture, designed to incorporate variational Bayesian inference at the encoder level. Specifically, all standard convolutional layers within the Residual U-blocks (RSUs) are replaced by Bayes by Backprop convolutional layers (BBBConv2d), which represent each filter weight as a learnable probability distribution rather than a fixed value. This modification introduces stochasticity into the forward pass through posterior sampling, acting as an implicit regularizer during training.

The proposed model preserves the nested U-structure of the original U²-Net, maintaining its ability to perform multi-scale feature fusion and precise boundary localization. The decoder architecture remains intact, allowing for seamless integration with the original skip connections and ensuring consistent output resolution.

By incorporating weight uncertainty into both the encoder and decoder via variational Bayesian convolutions, the model aims to enhance segmentation robustness under conditions of limited data and visual ambiguity. Furthermore, it provides a means for estimating epistemic uncertainty, which may offer potential value in corrosion detection scenarios by contributing to downstream risk assessments and decision-making processes in safety-sensitive applications.

This section is organized as follows:

- Section **5.2.1** introduces the U²-Net backbone and its hierarchical architecture.
- Section **5.2.2** describes how Bayesian convolutions are integrated into this structure.
- Section **5.2.3** presents the mathematical formulation of variational Bayesian layers.

5.2.1. Residual U-Blocks

U²-Net is used as the base architecture of our model because it can extract features at different scales and keep boundary details well. These abilities are important for segmenting corrosion areas, which often have irregular shapes, low contrast, and noisy edges. A key part of U²-Net is the Residual U-block (RSU), which is a small encoder–decoder unit placed at each level of the network [30].

Each RSU has its own U-shaped structure with downsampling and upsampling layers. This helps the network learn both global and local features. Residual connections inside the RSU also help keep training stable and make learning faster. Because of this, RSUs work well for corrosion images, which need both large-scale and small-scale information.

In our model, we retain the overall structure of RSUs but replace all standard convolutional layers within them with variational Bayesian convolutional layers (BBBConv2d). This modification improves generalization under data-scarce and visually ambiguous conditions.

The internal structure of the RSU and the location of the modified layers are illustrated in Fig. 11. Here, X denotes the input feature map and Y the output after RSU processing, including the skip connection. All convolutional layers marked in red are replaced with BBBConv2d in the proposed Bayesian-U²-Net.The next section explains this change in more detail.

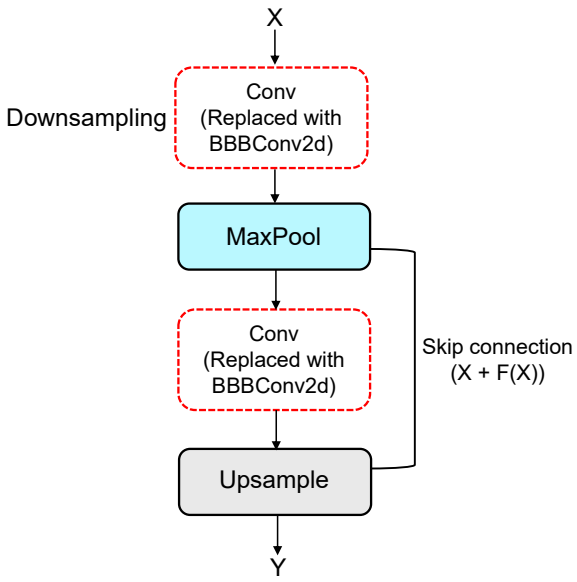


Fig. 11. Internal structure of the Residual U-block (RSU) used in U²-Net.

5.2.2. Bayesian Integration

The architecture of the proposed Bayesian-U²-Net is shown in Fig. 12. It follows the original nested U-shaped structure of U²-Net but replaces all standard convolutional layers in the Residual U-blocks (RSUs) with Bayesian convolutional layers (BBBConv2d). These Bayesian layers are based on the Bayes by Backprop method. They let the network learn distributions over weights and biases. This adds controlled randomness to the segmentation process without changing the network layout or connections.

The network starts with an input RGB image. The image passes through five encoder stages (En_1 to En_5). Each encoder uses an RSU module. These modules grow deeper at each stage (for example, RSU-7 in En_1 and RSU-6 in En_2). MaxPooling layers reduce the spatial size and increase the receptive field. All convolutions inside these RSUs are replaced by BBBConv2d to add uncertainty modeling while extracting features.

The decoder stages (De_5 to De_1) follow the same structure in reverse. Each decoder begins with an upsampling step, then uses an RSU with the same depth as its matching encoder. Skip

connections link each encoder stage to its corresponding decoder. These connections help keep spatial details and combine features from different levels. The decoder RSUs also use BBBConv2d to keep the Bayesian setup consistent across the whole network.

The model keeps U²-Net's original side-output design. Each decoder stage makes a prediction map. These side outputs are upsampled to match the input size. Then they are combined and passed through a final fusion layer to create the final binary segmentation mask.

The network outputs a pixel-wise corrosion mask. Each pixel shows whether corrosion is detected. Because all RSUs use Bayesian convolutions, the model changes from a fixed-output network to a probabilistic one. It can now show both the segmentation result and how confident it is in each prediction. This helps support reliable corrosion analysis in safety-critical tasks.

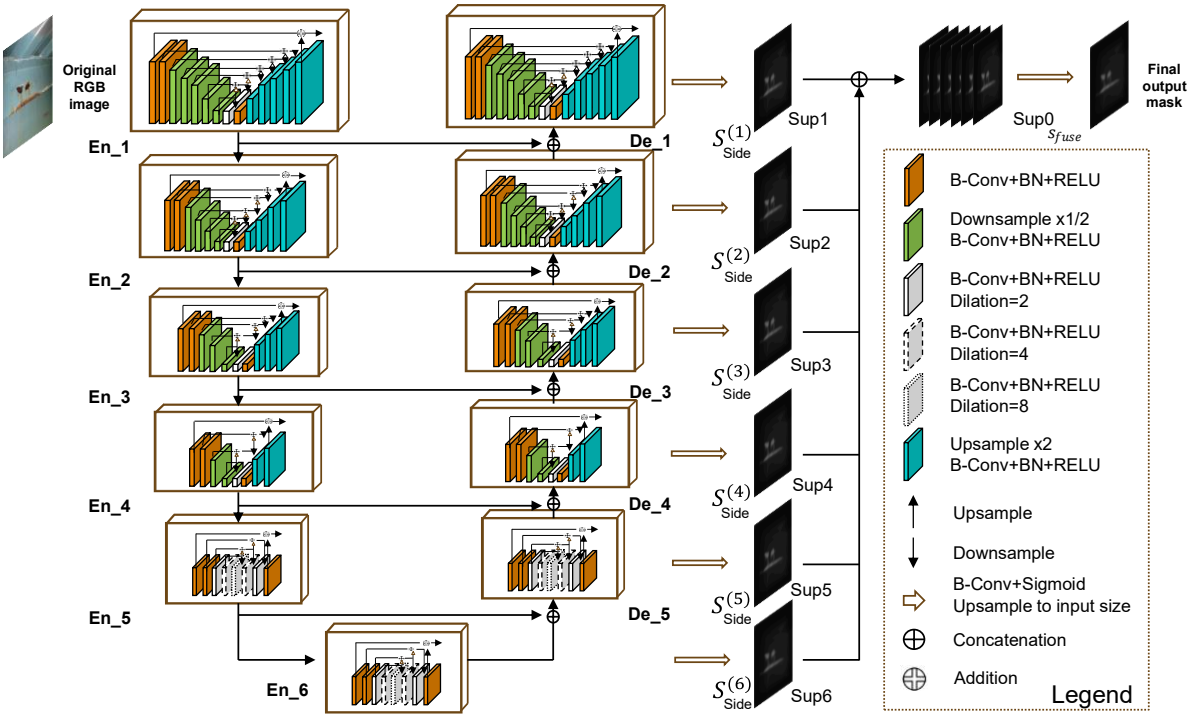


Fig. 12. Architecture of the proposed Bayesian-U²-Net.

5.2.3. Inference and Uncertainty

In this section, we first describe the variational modeling of convolutional weights using Bayes by Backprop. We then detail how this formulation supports stochastic forward passes and Monte Carlo inference. Finally, we present the corresponding loss function and training strategy that integrates segmentation accuracy and Bayesian regularization. This step-by-step exposition aligns with the computational flow of the Bayesian-U²-Net architecture during training and inference.

Bayesian Parameterization of Convolutional Layers

In the proposed Bayesian-U²-Net, all convolutional layers within the RSU modules, including those in both the encoder and decoder stages, are replaced with variational Bayesian convolutions based on the Bayes by Backprop framework, as introduced by Blundell et al [31]. Each convolutional weight w is modeled as a random variable following a Gaussian posterior distribution.

$$q(w|\theta) = \mathcal{N}(\mu, \sigma^2), \text{where} \sigma = \log\,(1 + \exp\,(\rho))$$

Here, μ and ρ are learnable parameters. The softplus function ensures that σ remains positive. Bias terms are modeled in a similar manner. This formulation allows the network to learn a posterior distribution over weights, introducing structured stochasticity into the segmentation process. Such stochasticity is particularly beneficial in corrosion segmentation tasks, where visual ambiguity, irregular textures, and class imbalance are commonly encountered.

Reparameterized Forward Computation

Variational inference is implemented in a differentiable manner by applying the reparameterization trick during training. This technique enables stochastic sampling of

convolutional weights while preserving the flow of gradients. Specifically, each weight w is sampled from a learned Gaussian distribution as follows:

$$w = \mu + \sigma \cdot \epsilon, \quad where\ \epsilon \sim \mathcal{N}(0,1)$$

This formulation transforms the sampling process into a deterministic function of the parameters μ and σ, along with a random noise term ϵ, thus allowing gradient-based optimization through stochastic layers. One sample is drawn per mini-batch during each forward pass and shared across all convolution operations.

Uncertainty-Aware Inference via Monte Carlo Sampling

Prediction confidence and epistemic uncertainty are estimated during inference using a Monte Carlo (MC) sampling strategy that exploits the stochastic nature of the Bayesian convolutional layers. The model remains in training mode to ensure stochastic sampling behavior is preserved. Each input image undergoes *T = 20* forward passes, with weights randomly sampled from the learned posterior distribution in each pass.

During inference, two types of pixel-level outputs are generated to support interpretability. The confidence map is computed by applying a sigmoid activation to the final logit output at each pixel:

$$P_{\text{conf}}(x) = \frac{1}{1 + \exp\left(-f(x)\right)}$$

Where $f(x)$ denotes the output logit for pixel x, and $P_{\text{conf}}(x)$ reflects the model's belief in the presence of corrosion at that location. In parallel, the epistemic uncertainty map is estimated using the pixel-wise variance of the sigmoid outputs across all stochastic forward passes:

$$\text{Uncertainty}(x) = \frac{1}{T}\sum_{t=1}^{T} f(x; w_t)^2 - \left(\frac{1}{T}\sum_{t=1}^{T} f(x; w_t)\right)^2$$

Here, $f(x; w_t)$ denotes the sigmoid output at the t-th Monte Carlo sample. High variance indicates prediction instability caused by limited data or ambiguous visual cues.

Loss Function with KL Divergence Regularization

The training objective of the Bayesian-U²-Net integrates both segmentation accuracy and posterior regularization. The total loss is composed of two terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}} + \beta \cdot \mathcal{L}_{\text{KL}}$$

Where $\mathcal{L}_{\text{seg}}$ is a hybrid segmentation loss, and $\mathcal{L}_{\text{KL}}$ is the Kullback–Leibler divergence that regularizes the posterior distributions of the Bayesian layers.

To balance pixel-level classification and region-level overlap, we define the segmentation loss as the sum of binary cross-entropy (BCE) and Dice loss:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{Dice}}$$

The BCE term improves per-pixel accuracy, while the Dice loss mitigates class imbalance and encourages consistent region segmentation. The KL divergence term enforces proximity between the learned weight posterior $q(w_i|\mu_i, \sigma_i^2)$ and a standard Gaussian prior $p(w_i) = \mathcal{N}(0,1)$, summed over all variational Bayesian layers:

$$\mathcal{L}_{\text{KL}} = \sum_i \text{KL}[q(w_i|\mu_i, \sigma_i^2) \parallel \mathcal{N}(0,1)]$$

To stabilize training in early epochs, the KL coefficient β is linearly annealed from 0 to 0.1 over the first 20 epochs. This allows the network to focus initially on learning effective feature representations before gradually incorporating posterior regularization.

5.3. Implementation and Inference

5.3.1. Structural Overview

Bayesian-U²-Net adopts a symmetrical encoder–decoder architecture based on the U²-Net backbone [30], where each stage is built with a Residual U-block (RSU). Each RSU contains a nested U-shaped substructure with internal skip connections, enabling efficient multi-scale feature extraction at limited depth.

To align with lightweight deployment and epistemic modeling goals, the input image is resized to $\mathbb{R}^{3\times320\times320}$. The encoder path includes six RSU modules (RSU7 to RSU4F), each followed by max pooling to halve spatial resolution while increasing channel depth. This results in a compressed representation of size $\mathbb{R}^{1\times512\times20\times20}$ at the bottleneck.

The decoder mirrors this structure in reverse order. At each stage, bilinearly upsampled features (using align_corners=True) are concatenated with skip-connected encoder outputs of matching spatial resolution. For instance, decoder stage 5d fuses features from stages 6 and 5, producing a tensor of size $\mathbb{R}^{1\times1536\times20\times20}$. This process continues symmetrically, restoring full resolution at $\mathbb{R}^{1\times64\times320\times320}$, followed by a 1×1 convolution to generate the final segmentation output $\mathbb{R}^{1\times1\times320\times320}$.

The RSU backbone is retained without structural modification to maintain compatibility with the original spatial feature flow. This design enables direct replacement of convolutional layers with Bayesian modules in the encoder, allowing variational sampling and epistemic uncertainty to propagate forward through skip connections. A detailed summary of layer configurations and feature dimensions is provided in Table 5.

Table 6. Stage-Wise Structural and Feature Map Configuration of Bayesian-U²-Net

Stage ID	Path	RSU Type	Internal Depth	Input Channels	Output Channels	Feature Size
stage1	Encoder	RSU7	7	3	64	320×320
Stage2	Encoder	RSU6	6	64	128	160×160
Stage3	Encoder	RSU5	5	128	256	80×80
Stage4	Encoder	RSU4	4	256	512	40×40
Stage5	Encoder	RSU4	4	512	1024	20×20
Stage6	Encoder	RSU4F	4F	1024	512	20×20
Stage5d	Decoder	RSU4	4	1536	512	40×40
Stage4d	Decoder	RSU4	4	1024	256	80×80
Stage3d	Decoder	RSU5	5	512	128	160×160
Stage2d	Decoder	RSU6	6	256	64	320×320
stage1d	Decoder	RSU7	7	128	64	320×320
output	Decoder	Conv1×1	–	64	1	320×320

The table includes RSU types, internal depth, input/output channels, and spatial resolution of feature maps across all encoder and decoder stages.

5.3.2. Training Configuration

The Bayesian-U²-Net was implemented using PyTorch and trained in an NVIDIA RTX GPU environment. Input images were resized to 320×320 and normalized to the [0,1] range, ensuring consistent spatial encoding and batch compatibility. Training employed the Adam optimizer with an initial learning rate of 0.001, selected through preliminary trials to balance convergence speed and stability under variational constraints. A ReduceLROnPlateau scheduler (patience: 5 epochs) was applied to reduce the learning rate upon validation loss stagnation, supporting fine-tuning and overfitting control during later stages.

The batch size was set to 12 after memory-performance benchmarking, offering a balance between computational feasibility and statistical reliability in stochastic weight sampling. Validation used a batch size of 1 to minimize memory usage and simulate real-world single-image inference, as in UAV-based corrosion inspections. Weights were initialized via Xavier uniform distribution to preserve activation scale and gradient flow, which is especially critical in architectures with deep skip connections and variational units. Training proceeded for 100 epochs, a value chosen to provide adequate learning without incurring performance degradation from overfitting.

The segmentation objective combined binary cross-entropy (BCE) with Dice loss, jointly addressing pixel-level classification and region-level overlap. For Bayesian training, a Kullback–Leibler (KL) divergence term was included to regularize the posterior. A linearly annealed coefficient $\beta \in [0, 0.1]$ was applied across the first 20 epochs to gradually activate the KL constraint, allowing the model to prioritize learning meaningful features before enforcing prior conformity.

During inference, epistemic uncertainty was estimated through Monte Carlo (MC) sampling. Each input image underwent $T = 20$ stochastic forward passes; the prediction mean served as the final segmentation mask, while the pixel-wise variance quantified uncertainty. This unified framework thus supports both deterministic outputs and uncertainty-aware diagnostics.

5.3.3. Deployment on Raspberry Pi

The feasibility of deploying Bayesian-U²-Net on resource-constrained platforms was evaluated using a Raspberry Pi 4 Model B (8 GB RAM), running the official 64-bit Raspberry Pi OS in a CPU-only environment. All inference tasks were performed using PyTorch in FP32 precision, without GPU support or hardware acceleration. The Raspberry Pi was selected based on its

relevance to embedded applications, including UAV-based bridge inspection and low-cost field robotics.

The model was exported using PyTorch’s "torch.save()" function and deployed directly in Python format, without conversion to "TorchScript" or application of quantization techniques. Despite using full-precision weights, the model file size remained small, approximately 78 MB, which allowed for efficient loading and execution on SD card-based storage systems. The deployment process was completed successfully, with no runtime errors or memory overflows observed.

Inference was conducted on 320×320 resolution images, consistent with the input resolution used during training. In deterministic mode, each forward pass took less than 50 milliseconds on average, as measured using Python’s "time" module. When performing 20 Monte Carlo samples for uncertainty estimation, the total latency increased linearly but stayed within acceptable limits for offline or near-real-time tasks. Peak memory usage was approximately 1.3 GB, as monitored using standard Linux tools such as "top".

Input images underwent preprocessing on-device using the illumination normalization pipeline described in Section 4, which combined GLF and LIME methods. This module was implemented in Python and introduced an additional delay of 15 to 20 milliseconds per image. The entire pipeline, including preprocessing, inference, and uncertainty estimation, was confirmed to be executable within a CPU-only setup.

Although limited by computational capacity, the current deployment demonstrates the practical viability of applying Bayesian segmentation models to embedded systems. Further improvements in speed and efficiency could be achieved through model quantization,

conversion to "TorchScript" or "ONNX" formats, or deployment on hardware-accelerated edge platforms such as Jetson Nano.

Fig. 13. Real-world deployment of Bayesian-U²-Net on Raspberry Pi 4 with Camera Module V3 and custom lens, enabling real-time corrosion capture and onboard inference.

5.4. Baseline Model

This section introduces the comparative baseline models used to evaluate the effectiveness of the proposed Bayesian-U²-Net. These models are not architectural variants of the proposed network but instead represent distinct segmentation paradigms, including both classical machine learning and modern deep learning approaches.

5.4.1. SVM Baseline

A Support Vector Machine (SVM) classifier was implemented to provide a traditional reference alongside the deep segmentation models [130]. Handcrafted feature extraction routines were developed in MATLAB, and the classifier operates at the pixel level. Each pixel is represented by a manually constructed feature vector derived from multiple visual cues.

Given an RGB image $I \in \mathbb{R}^{H\times W\times 3}$, a set of features are extracted per pixel using the following modules: Color features: RGB and HSV values computed from *colourget.m* and *hsvget.m*, represented as

$$f_{ij}^{\text{color}} = [R_{ij}, G_{ij}, B_{ij}, H_{ij}, S_{ij}, V_{ij}]$$

Gradient-based features: local grayscale mean μ_{ij}, standard deviation σ_{ij} ,gradient magnitude $|\nabla G|$,and orientation θ , obtained via *gradientchange.m* and *orientationangle.m*, expressed as

$$f_{ij}^{\text{grad}} = [\mu_{ij}, \sigma_{ij}, |\nabla G|_{ij}, \theta_{ij}]$$

These features are concatenated into a 10-dimensional vector for each pixel:

$$x_{ij} = \left[f_{ij}^{\text{color}}, f_{ij}^{\text{grad}}\right] \in \mathbb{R}^{10}$$

The training data was created by manually cropping image regions from corrosion and non-corrosion areas. From these regions, pixel-level feature vectors were extracted and assembled into a training matrix $X_{\text{train}} \in \mathbb{R}^{N\times 10}$, with binary labels $Y_{\text{train}} \in \{0,1\}^N$ A linear SVM model was trained using MATLAB's Classification Learner Toolbox.

During inference, the entire test image is converted to grayscale and features are computed pixel-wise using the same modules. The test feature matrix is constructed as $X_{\text{test}} \in \mathbb{R}^{HW\times 10}$, where H and W are the image height and width. Classification is then performed using the trained model:

$$\hat{Y}_{\text{flat}} = f_{\text{SVM}}(X_{\text{test}})$$

The result is reshaped to recover the original image geometry:

$$\hat{Y}_{\text{mask}} \in \mathbb{R}^{H\times W}$$

This approach, though lacking multiscale context or end-to-end representation learning, provides a classical benchmark for evaluating the gains offered by deep neural architectures such as U-Net and Bayesian-U²-Net.

5.4.2. U-Net Baseline

U-Net is a widely adopted encoder–decoder architecture for semantic segmentation, known for its simplicity and effectiveness in various domains [131]. In this study, we implement a five-level U-Net variant, consisting of symmetric encoding and decoding paths, as shown in Figure 4.5. The network takes an RGB input $x \in \mathbb{R}^{3\times320\times320}$ and progressively transforms it through a series of DoubleConv and pooling operations to extract hierarchical features.

Each encoder stage consists of a DoubleConv block, i.e., two sequential 3×33 convolutions with Batch Normalization and ReLU activation:

$$x' = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}_{3\times3}(\cdot)))$$

Downsampling is performed using 2×2 max pooling after each stage, reducing the spatial dimensions by half while increasing the number of channels. At the deepest layer (bottleneck), the feature representation reaches a dimension of 512×20×20. The decoder mirrors the encoder structure, using bilinear upsampling followed by concatenation with the corresponding encoder features (skip connections):

and ultimately reaches a bottleneck representation $x_5 \in \mathbb{R}^{512\times20\times20}$. The decoder then applies bilinear upsampling and concatenation:

$$x_{\mathrm{dec}_i} = \mathrm{DoubleConv}([\mathrm{Up}(x_{\mathrm{dec}_{i+1}}) \parallel x_{\mathrm{enc}_i}])$$

where ‖ denotes channel-wise concatenation. For example, the first decoder stage (up1) receives $x_{\mathrm{dec}_2} \in \mathbb{R}^{512\times20\times20}$ and concatenates it with the encoder output $x_{\mathrm{enc}_4} \in \mathbb{R}^{512\times20\times20}$, forming a tensor of size 1024×20×20, which is then reduced to 256 channels through DoubleConv.

The channel transitions across the network follow this pattern:

$$3 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 512 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 64 \rightarrow 1$$

Finally, a 1×1 convolution is applied:

$$\hat{y} = \text{Conv}_{1\times1}(x_{\text{final}}), \hat{y} \in \mathbb{R}^{1\times320\times320}$$

to map the 64-channel feature map to a binary segmentation mask. The entire channel and resolution flow is illustrated in Fig. 14.

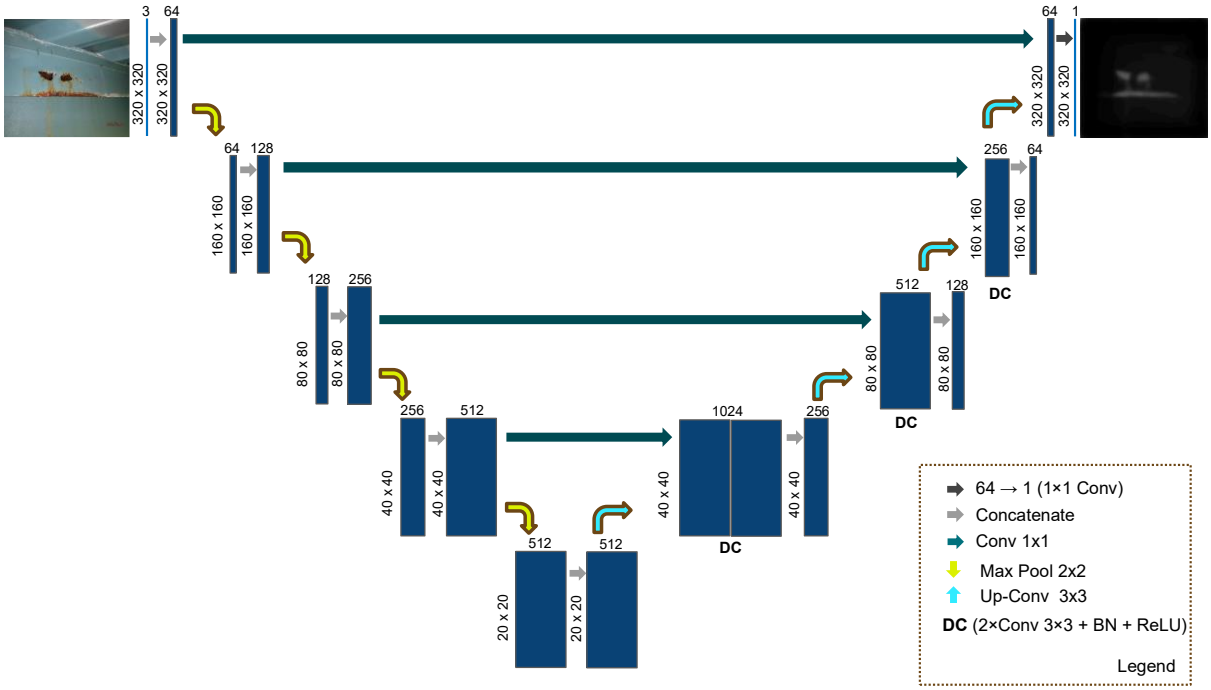


Fig. 14. U-Net baseline with a symmetric encoder–decoder structure, five encoding and four decoding stages, skip connections, DoubleConv blocks (2×Conv3×3 + BN + ReLU), and a final 1×1 convolution output layer.

6. Chapter 6 –Results and Discussion

6.1. Evaluation Metrics

The proposed corrosion segmentation framework was evaluated using three metrics: Recall, F1 score, and Intersection over Union (IoU). These metrics collectively reflect classification accuracy and spatial overlap between predicted masks and ground truth annotations. All evaluations are based on four pixel-level outcomes: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP refers to correctly predicted corroded pixels, TN to correctly identified non-corroded pixels, FP to non-corroded pixels incorrectly classified as corroded, and FN to corroded pixels that the model failed to detect.

The Recall metric measures the proportion of actual corrosion pixels correctly identified, and is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall is particularly important in safety-critical applications such as corrosion detection, where missed defects can have severe consequences.

The F1 Score, which balances recall and false positive rate, is defined as:

$$\text{F1Score} = 2 \times \frac{TP}{2TP + FP + FN}$$

This formulation eliminates the need to explicitly reference precision, while still providing a balanced view of detection accuracy, especially under class imbalance where corroded pixels represent only a small portion of the image.

The Intersection over Union (IoU) was utilized to quantify the spatial overlap between predicted segmentation masks and ground truth annotations. It is defined as:

$$\text{IoU} = \frac{TP}{TP + FP + FN}$$

A higher IoU value indicates a greater degree of correspondence between the predicted corrosion regions and the manually annotated labels, which is critical for ensuring reliability in downstream maintenance applications.

Furthermore, the training loss was monitored throughout the model development process to capture the optimization dynamics. The cross-entropy loss, which is a standard choice for semantic segmentation tasks, was adopted and computed as follows:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\left(p_{i,c}\right)$$

Where N is the number of pixels, C is the number of classes, $y_{i,c}$ denotes the ground truth label for pixel i and class c, and $p_{i,c}$ represents the predicted probability for the same. Monitoring the loss over epochs provides insights into the convergence behavior, potential overfitting, and generalization ability of the model.

6.2. Quantitative Results

This section presents the quantitative evaluation of three semantic segmentation models: U-Net, U²-Net, and the proposed BU²-Net. All models were trained across 45 independent runs, each consisting of 100 epochs. Performance on the validation set was measured using three standard metrics: F1 score, Intersection over Union (IoU), and recall. In addition to accuracy, training efficiency was assessed by analyzing the loss convergence trends and the average runtime per epoch. A statistical analysis was subsequently conducted using independent-

sample *t*-tests and effect size calculations (Cohen's *d*) to determine the significance and practical relevance of the observed differences.

6.2.1. Accuracy Metrics Evaluation

The segmentation performance of U-Net, U²-Net, and BU²-Net was evaluated on the validation set using F1-score, Intersection over Union (IoU), and recall. As illustrated in Fig. 15, each bar represents the average result over 45 independent runs, with error bars indicating the standard deviation. BU²-Net demonstrates consistently higher scores across all three metrics.

Specifically, BU²-Net achieves an average F1-score of 75.088 percent with a standard deviation of 2.390, outperforming U-Net at 69.590 with a deviation of 4.242 and U²-Net at 72.437 with a deviation of 3.544. For IoU, BU²-Net reaches 60.350 percent, slightly higher than U-Net at 60.195 and notably higher than U²-Net at 57.101. The standard deviations are 3.017 for BU²-Net, 3.356 for U-Net, and 4.334 for U²-Net. In terms of recall, BU²-Net records 71.537 percent with a deviation of 3.200, compared to 66.456 for U-Net and 68.232 for U²-Net, with standard deviations of 3.857 and 5.060 respectively.

Compared to U-Net, BU²-Net achieves absolute gains of approximately 7.9 percentage points in F1-score and 7.6 in recall, along with a small improvement of 0.2 in IoU. Similar improvements are observed relative to U²-Net, with the largest margin again in recall.

In addition to higher average values, BU²-Net exhibits the lowest standard deviations among the models, indicating more consistent performance across runs. The error bars for F1-score and recall are clearly separated, highlighting distinct performance differences. In contrast, the overlap in IoU error bars between BU²-Net and U-Net suggests a smaller margin in this metric.

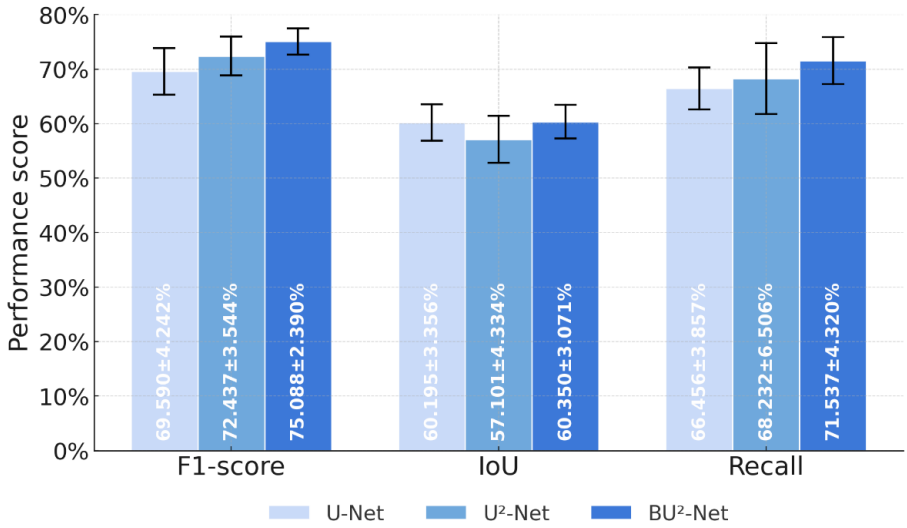


Fig. 15. Performance comparison of U-Net, U²-Net, and BU²-Net on the validation set across F1-score, IoU, and recall. Error bars indicate standard deviation over 45 runs.

The performance distributions of U-Net, U²-Net, and BU²-Net on the validation set are compared using boxplots across F1-score, Intersection over Union (IoU), and recall metrics, as shown in Fig. 16. The left column illustrates the distribution of final-epoch scores from 45 independent training runs, while the right column presents the distribution of per-run average scores computed over all 100 epochs.

Among the three models, U-Net exhibits the lowest median scores across all metrics, with relatively narrow interquartile ranges (IQR), indicating modest but consistent performance. U²-Net reaches higher maximum values in some cases, but its distributions are notably wider, with a larger IQR and more outliers, particularly in the recall metric. This reflects considerable variability and instability across different training seeds. In contrast, BU²-Net consistently achieves the highest medians and displays the most compact distributions, especially in the mean performance plots, suggesting more stable and concentrated performance.

BU²-Net also shows the fewest outliers across all metrics, indicating robustness to initialization variance. U²-Net, by comparison, contains more low-performing outliers, most notably in final-

epoch recall. The contrast between the left and right columns further highlights convergence dynamics: BU²-Net maintains high performance both at the final epoch and across the full training span, while U²-Net shows greater disparity between its final and average scores, implying potential late-epoch instability. Among the three metrics, recall shows the greatest dispersion between models, reinforcing the consistency advantage of BU²-Net in that aspect.

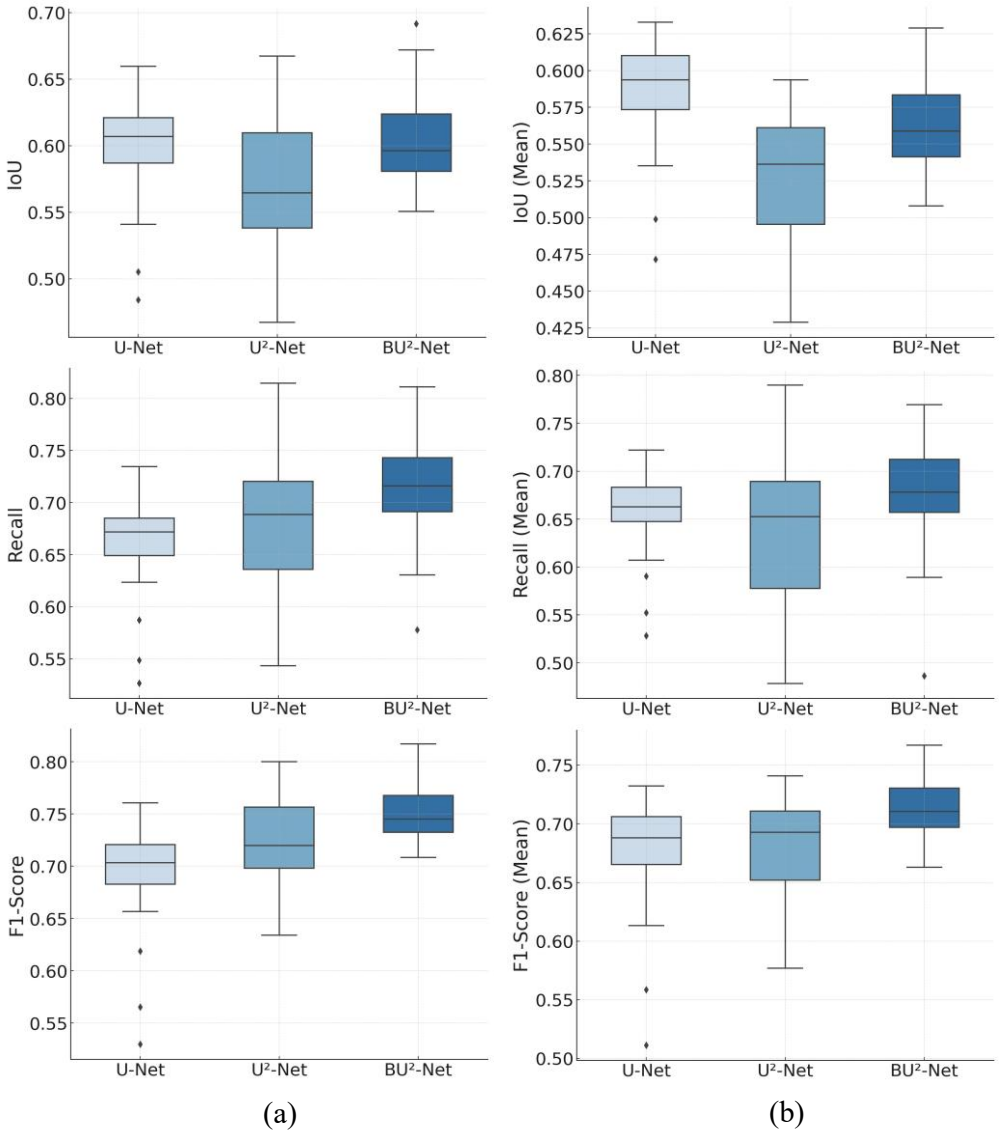


Fig. 16. Boxplot comparison of final-epoch (a) and mean-per-epoch (b) validation metrics for U-Net, U²-Net, and BU²-Net across 45 runs.

Fig. 16. Boxplot comparison of final-epoch (a) and mean-per-epoch (b) validation metrics for U-Net, U²-Net, and BU²-Net across 45 runs.

The F1 score progression over 100 training epochs for U-Net, BU²-Net, and U²-Net is shown in Fig. 17. Subplots (a)–(c) depict six representative training trajectories for each model: Max Final, Min Final, Median Final, Early Stable, Late Stable, and Most Volatile. Each subplot also overlays the epoch-wise mean F1 score and its corresponding standard deviation band across 45 independent training runs. Subplot (d) summarizes the mean F1 performance and standard deviation bands for all three models across the full training span.

During the initial 20 epochs, all models exhibit a rapid increase in F1 scores. BU²-Net demonstrates tightly grouped representative curves with early stabilization. Its average curve shows a sharp ascent followed by a steady high plateau. U²-Net shows large disparities among its six representative runs, especially in early epochs, with noticeable oscillations and delayed convergence. U-Net displays moderately spaced curves, with early-stage smooth improvement and stable convergence afterward.

In subplot (d), BU²-Net achieves the highest average F1 score across the full training range, followed by U-Net and then U²-Net. BU²-Net maintains the narrowest standard deviation band throughout training, indicating low variability across runs. U²-Net shows the broadest standard deviation, especially before epoch 40, while U-Net's deviation range lies between the two.

The six representative trajectories provide additional insight into model behavior. For BU²-Net, the Max Final run rapidly surpasses 0.85 in the first 10 epochs and maintains a high level throughout. The Min Final run exhibits slower ascent but remains stable. The Median Final curve closely matches the average trend, indicating consistent performance across runs. U²-Net shows the widest spread between Max and Min Final runs, with Early Stable and Late Stable runs diverging significantly in convergence timing. Its Most Volatile curve fluctuates

frequently and deviates from the mean. U-Net's Max and Min Final curves are moderately separated. Early and Late Stable runs converge steadily with minimal oscillation. The Most Volatile line for U-Net stays relatively close to the mean, showing controlled variability.

Standard deviation analysis further supports these patterns. BU²-Net's deviation band remains the tightest, reflecting highly consistent training dynamics. U²-Net's band is significantly wider, with notable variance across epochs. U-Net's standard deviation decreases steadily, reflecting improved consistency in later epochs.

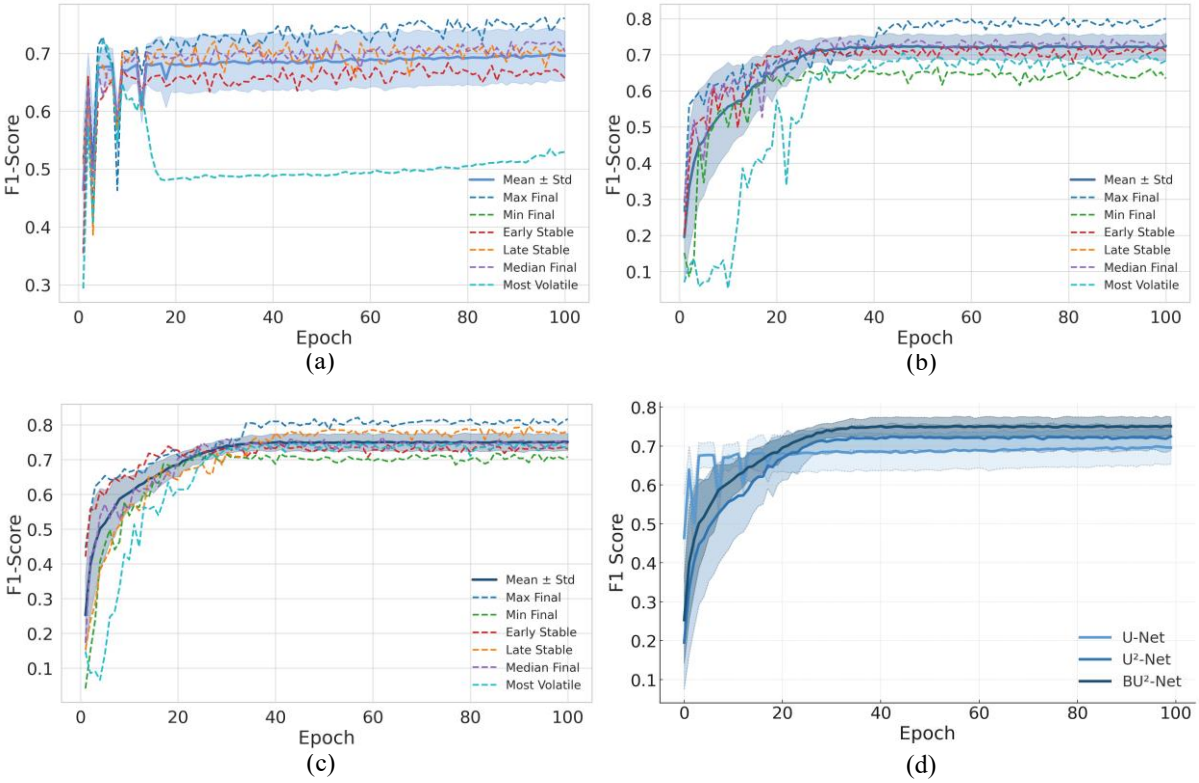


Fig. 17. Training F1 score curves over 100 epochs. (a) U-Net, (b) BU²-Net, and (c) U²-Net show six representative runs with average ± std; (d) compares mean and standard deviation across models.

The IoU performance over 100 training epochs for U-Net, BU²-Net, and U²-Net is compared in Fig. 18. Subplots (a)–(c) display six representative training runs for each model: Max Final,

Min Final, Median Final, Early Stable, Late Stable, and Most Volatile. The epoch-wise mean IoU and the associated standard deviation bands across 45 independent runs are overlaid. Subplot (d) summarizes the average IoU values and standard deviation ranges across the full training process for all three models.

During the first 20 to 30 epochs, all three models demonstrate a noticeable upward trajectory in IoU performance, eventually stabilizing as training proceeds. BU²-Net exhibits tightly grouped representative runs with a steadily increasing average IoU curve, suggesting strong convergence behavior. In contrast, U²-Net shows substantial variability across its six representative curves, particularly marked by pronounced early-stage fluctuations. U-Net lies between the two, with moderately dispersed curves that feature smooth early growth and eventual stabilization.

Subplot (d) further highlights these patterns. BU²-Net maintains the highest average IoU across epochs, while U-Net ranks second, and U²-Net consistently trails behind. Notably, BU²-Net's standard deviation bands remain narrow throughout, indicating strong consistency across runs. U²-Net, on the other hand, exhibits the largest spread, especially during early epochs, while U-Net displays an intermediate level of variability.

Examining the six representative trajectories reveals deeper insights into training dynamics. BU²-Net’s Max Final curve achieves a high IoU within the initial epochs and continues to improve steadily. The Min Final curve progresses more gradually but remains stable throughout, and the Median Final curve aligns closely with the overall mean. In contrast, U²-Net’s trajectories vary widely: Max and Min Final differ significantly, Early Stable and Late Stable follow distinct convergence paths, and the Most Volatile curve demonstrates pronounced instability across epochs. U-Net’s representative runs show moderate dispersion,

with both Early Stable and Late Stable curves converging gradually and consistently. Its Most Volatile curve, though fluctuating, largely adheres to the mean trend.

The analysis of standard deviation bands further substantiates these observations. BU²-Net sustains a consistently narrow deviation range, reflecting minimal sensitivity to initialization and training noise. U²-Net displays persistent and broader deviations, particularly prominent in the first 40 epochs, indicating a higher degree of instability. U-Net's deviations are moderate and gradually taper off, suggesting steady improvement in consistency over the course of training.

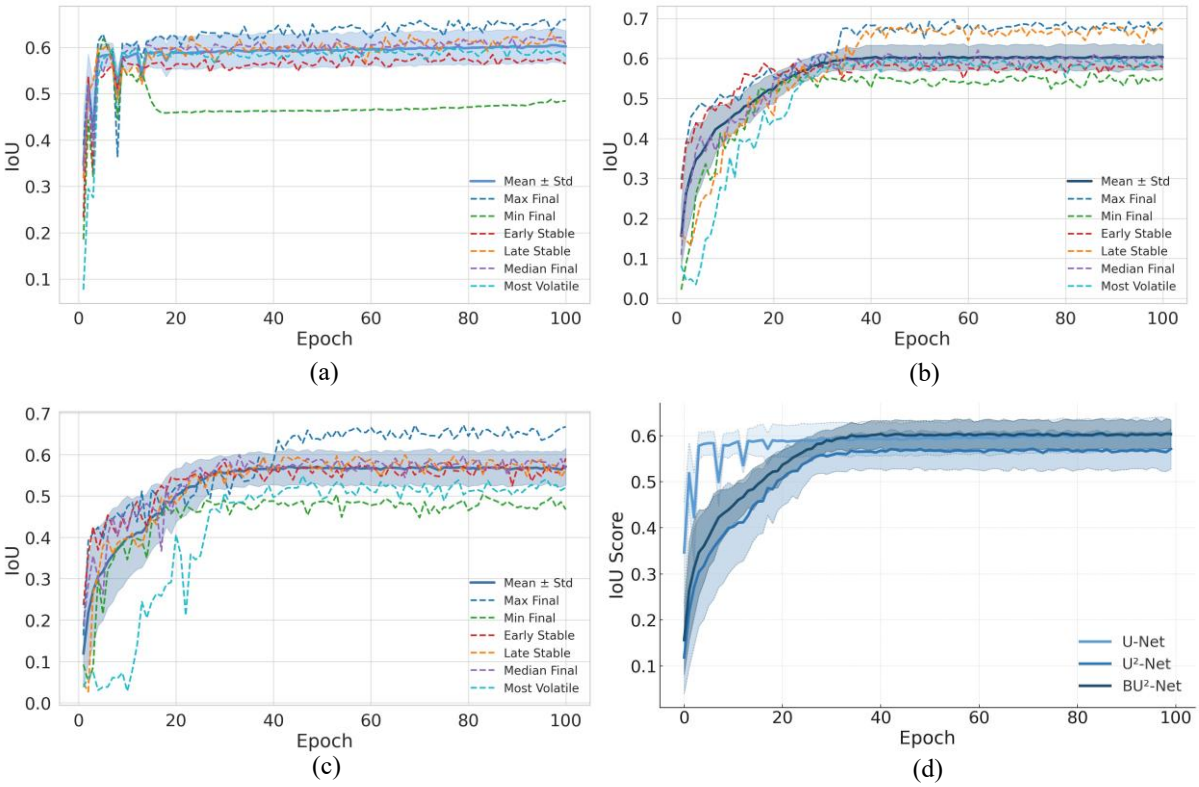


Fig. 18. Training IoU curves over 100 epochs. (a) U-Net, (b) BU²-Net, and (c) U²-Net show six representative runs with average ± std; (d) compares mean and standard deviation across models.

The Recall performance over 100 training epochs for U-Net, U²-Net, and BU²-Net is shown in Fig. 19. Subplots (a)–(c) correspond to the three models respectively and include six representative training trajectories: Max Final, Min Final, Median Final, Early Stable, Late Stable, and Most Volatile. Each subplot also includes the mean Recall curve from 45 independent runs, along with the corresponding standard deviation bands. Subplot (d) summarizes the epoch-wise mean Recall values and standard deviation ranges for all three models.

In the early training phase (approximately epochs 20 to 30), all three models demonstrate a clear upward trajectory in Recall performance before transitioning into a more stable stage. BU²-Net displays a compact clustering of its representative training curves, accompanied by a steadily ascending average Recall trend. In contrast, U²-Net reveals considerable variability across its six highlighted runs, marked by pronounced fluctuations in the early epochs. U-Net occupies a middle ground, with moderate dispersion and a relatively smooth progression toward convergence.

Subplot (d) provides a comparative view of mean Recall values across epochs. BU²-Net consistently achieves the highest average Recall, followed by U-Net, while U²-Net trails behind. The corresponding standard deviation bands shrink as training progresses, with BU²-Net maintaining the most confined range, U²-Net exhibiting the broadest spread, and U-Net positioned between the two.

A closer look at the six representative runs underscores these trends. BU²-Net's Max Final curve reaches a high Recall value rapidly within the first 10 epochs and continues on an upward trajectory. Its Min Final curve advances steadily, while the Median Final closely mirrors the overall average, reflecting stable and consistent performance. U²-Net's trajectories, on the other hand, are widely scattered: the gap between its Max and Min Final runs is substantial, Early

and Late Stable curves show divergent convergence behavior, and the Most Volatile trajectory fluctuates unpredictably throughout training. U-Net's representative curves demonstrate a more cohesive pattern, with both Early and Late Stable runs progressing smoothly and the Most Volatile curve remaining relatively aligned with the mean.

The behavior of the standard deviation bands further confirms these findings. BU²-Net quickly reaches a low-variance regime and maintains it consistently, indicating strong robustness. U²-Net experiences significant deviation in the early stages, particularly within the first 50 epochs, highlighting its sensitivity to initialization and stochasticity. U-Net presents a moderate deviation profile, with early fluctuations gradually giving way to more stable variance in the later epochs.

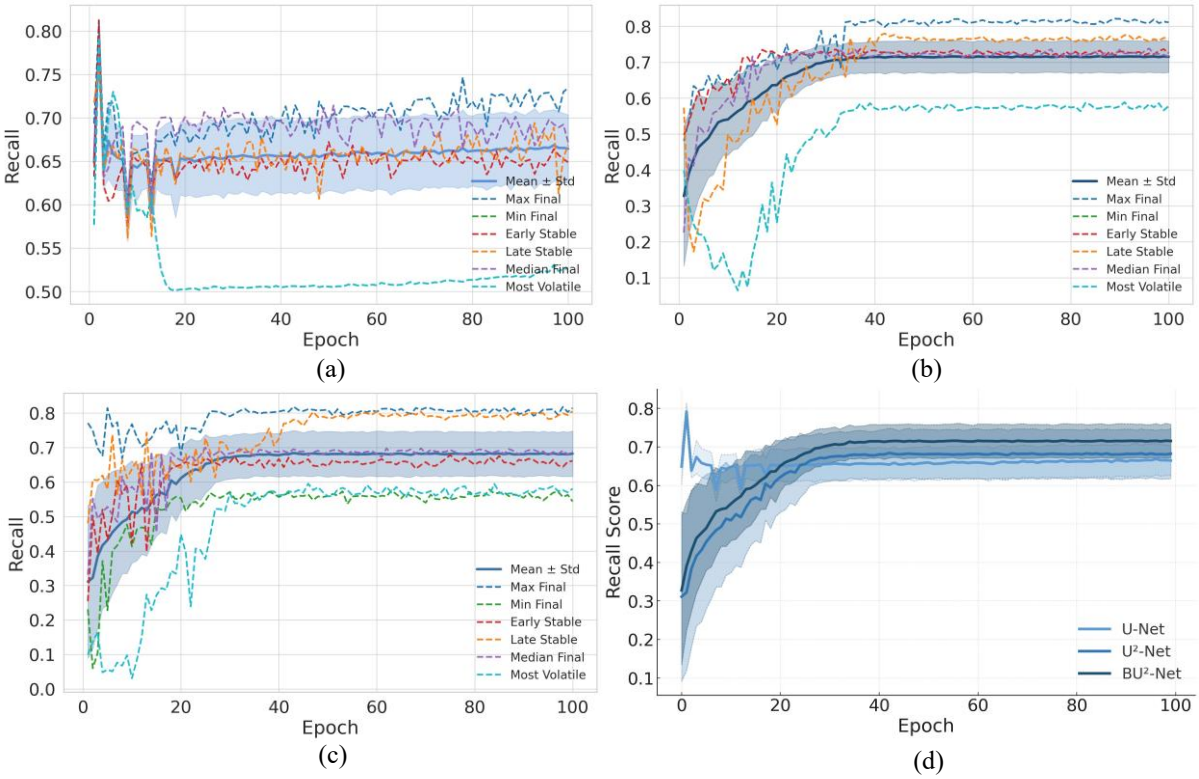


Fig. 19. Training Recall curves over 100 epochs. (a) U-Net, (b) BU²-Net, and (c) U²-Net show six representative runs with average ± std; (d) compares mean and standard deviation across models.

6.2.2. Training Efficiency

BU²-Net achieved the lowest average training loss at 1.128, outperforming U²-Net (1.298) and U-Net (1.323). It also exhibited the smallest standard deviation of 0.0189, indicating greater training stability and consistent convergence. The range of training loss for BU²-Net, from 1.079 to 1.165, was the narrowest among all models, further supporting its robustness. In comparison, U-Net showed the highest average loss and the widest fluctuation (1.217 to 1.409), suggesting greater variability across runs. U²-Net performed moderately but did not surpass BU²-Net on any loss-related metric. These descriptive statistics indicate that BU²-Net provides more reliable training behavior under identical experimental conditions.

The dual-axis training dynamics of U-Net (left), BU²-Net (middle), and U²-Net (right) across 100 epochs are illustrated in Fig. 20. In each subplot, solid lines represent training loss (left y-axis), while dashed lines indicate recall performance (right y-axis). Six representative training runs (Max Final, Min Final, Median Final, Early Stable, Late Stable, Most Volatile) are included for each model to provide a detailed comparison of convergence behavior and performance fluctuations.

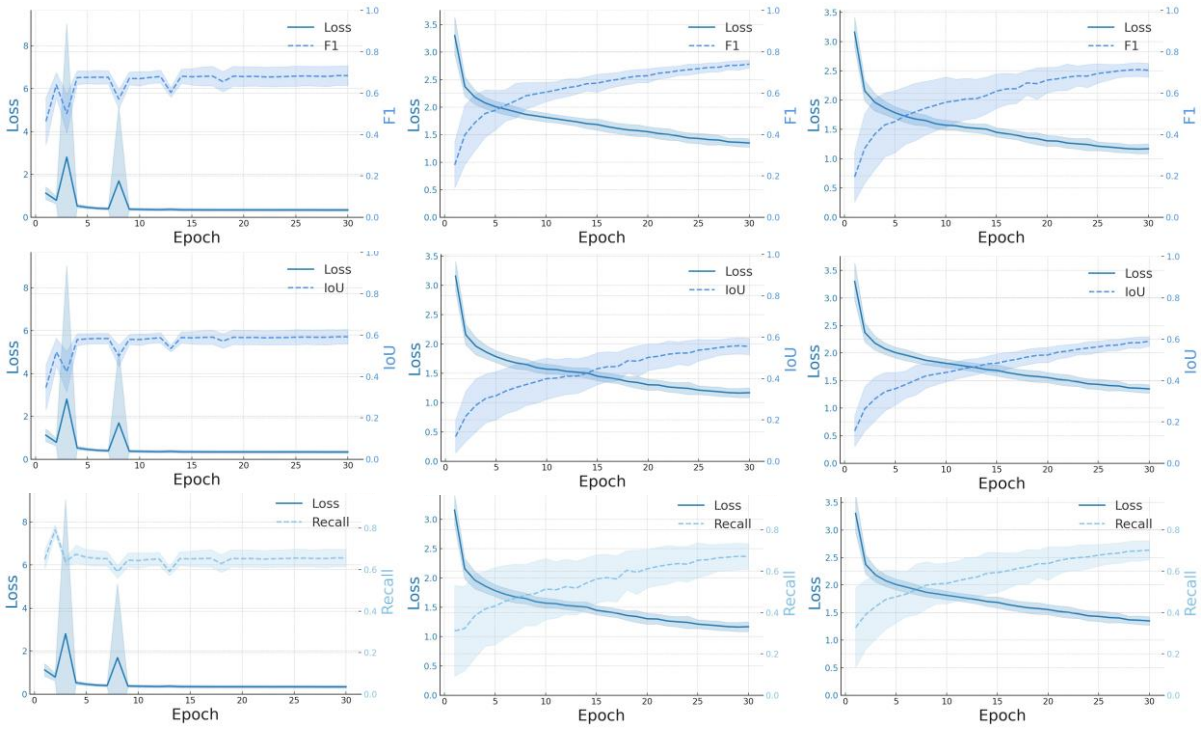
Across all models, the loss curves demonstrate a rapid decrease during the first 20–30 epochs, followed by gradual stabilization. However, the rate and consistency of convergence differ significantly. For BU²-Net, the loss trajectories of all six representative runs are tightly clustered and exhibit a smooth descent, aligning closely with a consistently rising recall curve. This synchronous evolution suggests a stable learning process where improved optimization directly contributes to enhanced recall accuracy.

In contrast, U-Net displays considerable dispersion in both loss and recall curves. The loss descent is marked by early-stage oscillations, particularly in the Most Volatile and Min Final runs. The corresponding recall lines vary widely in slope and amplitude, indicating inconsistent

learning behavior. For example, some runs show early recall spikes without sustained gain, while others lag behind despite steady loss minimization.

U²-Net's patterns lie between the two. While its loss curves are generally smooth, moderate variance appears in recall progression, especially in the Min Final and Most Volatile runs. The Early Stable and Late Stable runs demonstrate recall improvement that more closely tracks loss reduction, suggesting relatively predictable training behavior.

Further inspection reveals that BU²-Net consistently exhibits high alignment between recall gains and loss reduction across all six categories. U-Net frequently presents a misalignment, with instances where loss reduction does not lead to recall improvement or where recall fluctuates despite steady optimization. U²-Net shows a moderate level of alignment, with noticeable but not severe divergence in certain runs.



(b) (b) (c)

Fig. 20. Dual-axis visualization of loss (solid lines, left y-axis) and recall (dashed lines, right y-axis) across 100 training epochs for U-Net (left), BU²-Net (middle), and U²-Net (right), each showing six representative runs to highlight convergence and stab

The training time statistics for U-Net, BU²-Net, and U²-Net across 45 independent runs are shown in Fig. 21. Subplot (a) illustrates the average per-epoch training time, while subplot (b) summarizes the total training time for each full run.

In subplot (a), U-Net exhibits a periodic fluctuation in per-epoch training time, ranging from approximately 18 to 40 seconds. This oscillation persists throughout the 100 epochs, suggesting unstable computational behavior. BU²-Net maintains a relatively stable profile with an average of around 22 seconds per epoch and only minor variation. U²-Net records the lowest and most consistent per-epoch time, remaining just below 20 seconds across the entire training span.

Subplot (b) highlights the total runtime variability among the models. U-Net displays the widest spread in cumulative training time, with values ranging from roughly 2600 to over 4200 seconds. BU²-Net shows uniform runtimes centered around 2300 seconds across all runs. U²-Net again demonstrates the shortest total training time, typically under 2000 seconds, and minimal variation across the 45 runs.

Together, the two subplots reflect clear differences in training efficiency and time stability among the three models. U-Net incurs greater variability, while U²-Net achieves consistent and efficient runtimes under the same experimental settings.

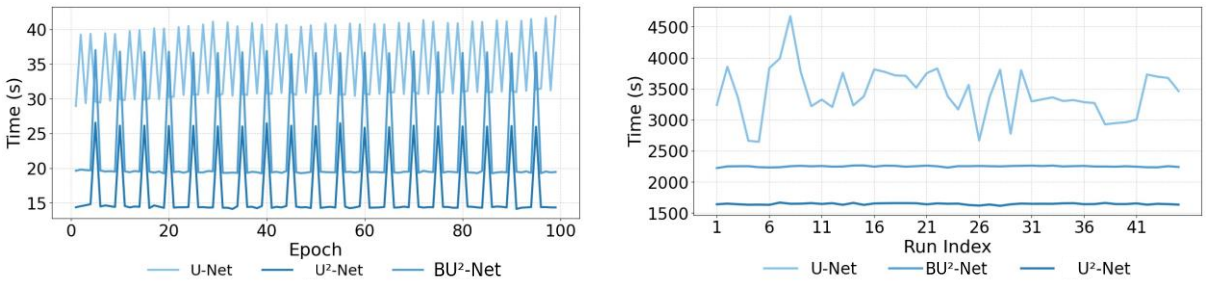


Fig. 21. Training time comparison of U-Net, U²-Net, and BU²-Net across 45 runs. (a) Per-epoch training time over 100 epochs. (b) Total training time per run.

The training time distribution and average runtime per model are presented in Fig. 22. Subplot (a) shows the total training time of U-Net, BU²-Net, and U²-Net across 45 runs using a boxplot, while subplot (b) displays the mean training time for each model accompanied by standard deviation error bars.

In subplot (a), U-Net exhibits a wide spread in training time, with values ranging from approximately 2600 to 4500 seconds. The interquartile range is large, and a clear outlier is present, indicating high variability in computational performance. BU²-Net demonstrates a narrow distribution centered around 2300 seconds, with minimal deviation across runs. U²-Net displays the most compact box, with training times consistently clustered near 1700 seconds and no visible outliers, suggesting a highly stable runtime profile.

Subplot (b) reinforces these observations by presenting the mean training time and variability for each model. U-Net shows the highest average time, exceeding 3400 seconds, along with the largest standard deviation. BU²-Net averages slightly above 2300 seconds with minimal fluctuation. U²-Net reports the lowest mean runtime, approximately 1700 seconds, with the smallest standard deviation among the three.

These results provide a comparative view of computational time characteristics, emphasizing the differences in runtime stability and per-run efficiency across the three model architectures.

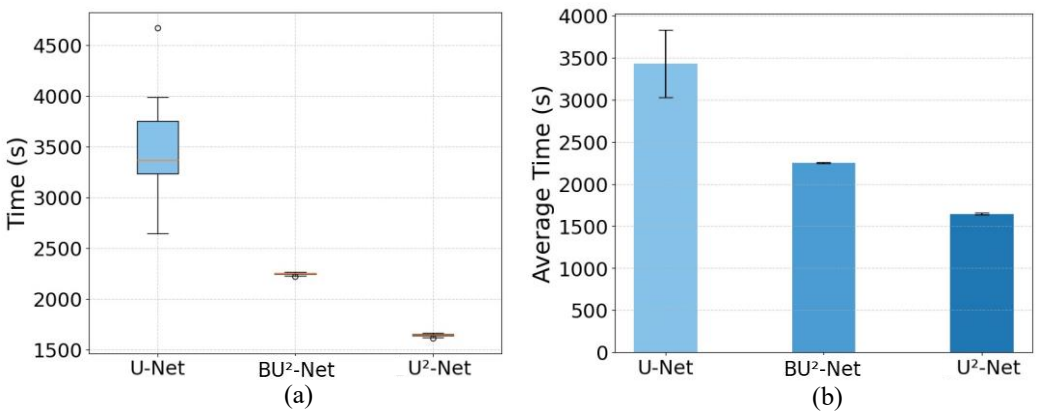


Fig. 22. Comparison of total training time across 45 runs. (a) Boxplot showing runtime distribution and variance. (b) Bar chart of average training time with standard deviation.

6.2.3. Statistical Comparison

The results of the t-test across 45 independent runs, as summarized in Table 6, confirm statistically significant differences among U-Net, U²-Net, and BU²-Net in four key metrics: F1-score, IoU, recall, and loss. BU²-Net achieves the highest mean F1-score at 0.751, clearly outperforming U²-Net at 0.724 and U-Net at 0.696. The associated p-values in both comparisons are below 0.001, indicating strong statistical significance. Similarly, BU²-Net attains a mean recall of 0.715, which is higher than U²-Net's 0.682 and U-Net's 0.665. All pairwise comparisons involving BU²-Net in this metric yield p-values under 0.01. In contrast, the recall difference between U-Net and U²-Net is not significant, with a p-value of 0.124, suggesting their performance distributions are statistically comparable.

The IoU results reveal a subtler pattern. BU²-Net and U-Net show nearly identical mean values, 0.603 and 0.602 respectively, with no significant difference between them. However, both models significantly outperform U²-Net, which reports a lower average of 0.571. The corresponding comparisons produce p-values well below 0.001, reflecting U²-Net's consistent underperformance in this area.

For the loss metric, BU²-Net exhibits improved optimization performance, with a mean loss of 1.128 compared to 1.298 for U²-Net and 1.323 for U-Net. These differences are statistically significant, as indicated by p-values below 1e-5 and large corresponding t-statistics. While BU²-Net shows relatively better predictive accuracy and training stability, U-Net maintains moderate performance with greater variance. U²-Net underperforms in several key metrics, suggesting areas where further refinement may be needed. The results highlight the potential benefits of Bayesian integration, though more extensive evaluations across diverse datasets would further validate these findings.

Table 7. T-test Results

Metric	Model A	Model B	Mean A	Mean B	T-statistic	P-value
F1	U-Net	U²-Net	0.696	0.724	–3.417256	0.000971
F1	U-Net	BU²-Net	0.696	0.751	–7.490353	0.000000
F1	U²-Net	BU²-Net	0.724	0.751	–4.112906	0.000097
IoU	U-Net	U²-Net	0.602	0.571	3.743998	0.000333
IoU	U-Net	BU²-Net	0.602	0.603	–0.225922	0.821790
IoU	U²-Net	BU²-Net	0.571	0.603	–4.057245	0.000116
Loss	U-Net	U²-Net	0.323	1.128	–49.496797	0.000000
Loss	U-Net	BU²-Net	0.323	1.298	–56.729489	0.000000
Loss	U²-Net	BU²-Net	1.298	1.128	–8.525060	0.000000
Recall	U-Net	U²-Net	0.665	0.682	–1.557737	0.123709
Recall	U-Net	BU²-Net	0.665	0.715	–5.819536	0.000000

Metric	Model A	Model B	Mean A	Mean B	T-statistic	P-value
Recall	U²-Net	BU²-Net	0.682	0.715	–2.807032	0.006340

Effect size comparisons using Cohen's d are summarized in Figure 23. The table covers four core metrics: F1-score, Intersection over Union, recall, and loss. These values represent the magnitude of pairwise differences across 45 training runs. Each comparison includes a 95 percent confidence interval. According to standard interpretation, an absolute value of Cohen's d greater than 0.8 typically indicates a large effect.

The largest differences appear in the loss metric. The comparison between U-Net and BU²-Net yields an effect size of negative 11.96, while the difference between U-Net and U²-Net reaches negative 10.43. Both results confirm substantial improvements in optimization. BU²-Net also outperforms U²-Net with an effect size of 1.80, reinforcing its consistent training efficiency across runs.

In classification performance, BU²-Net demonstrates clear advantages. For F1-score, the effect size reaches negative 1.58 compared to U-Net and negative 0.87 compared to U²-Net. The gap between U-Net and U²-Net is smaller, at negative 0.72. With respect to Intersection over Union, BU²-Net outperforms U²-Net by a margin of negative 0.86, and U-Net shows an advantage over U²-Net with a positive difference of 0.79. Recall comparisons also favor BU²-Net, with effect sizes of negative 1.23 versus U-Net and negative 0.59 versus U²-Net.

These values highlight the consistent superiority of BU²-Net in both segmentation accuracy and training stability. The magnitude of effects across multiple metrics confirms its robust and reliable performance.

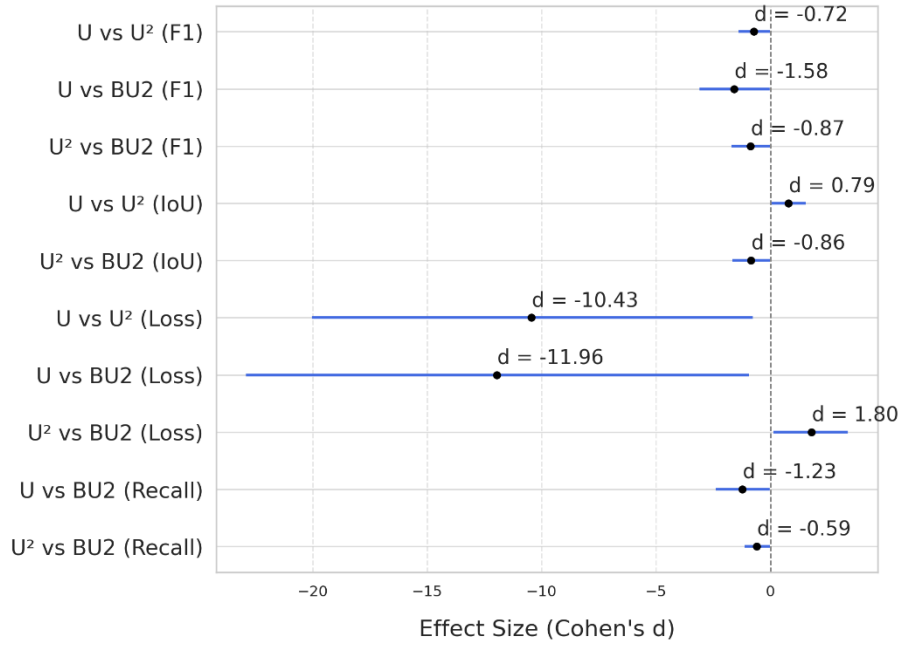


Fig. 23. Effect size (Cohen's d) comparison across models for F1-score, IoU, recall, and loss, with 95% confidence intervals.

6.2.4. Discussion

The comparative results across F1 score, Intersection over Union (IoU), and training loss consistently highlight the superior performance of the proposed Bayesian-U²-Net (BU²Net) relative to the baseline U-Net and U²-Net architectures. In all three metrics, BU²Net not only achieves higher central values, such as median F1 and IoU, but also demonstrates lower variance, fewer extreme outliers, and smoother training dynamics. These outcomes reflect improved robustness, better generalization, and more stable optimization, which are particularly important when working with visually ambiguous corrosion patterns or limited training data.

Comprehensive evaluation across multiple dimensions reveals clear performance and optimization differences among U-Net, U²-Net, and the proposed BU²-Net. BU²-Net consistently outperforms the other two models in segmentation accuracy, training stability, and computational efficiency.

In terms of performance metrics, BU²-Net achieves the highest average F1-score at 0.751, which is significantly higher than U²-Net at 0.724 and U-Net at 0.696. The corresponding p-values for these comparisons are all below 0.01, and the Cohen's d values fall within the medium-to-large effect range, indicating both statistical and practical significance. For recall, BU²-Net records an average of 0.715, exceeding U²-Net at 0.682 and U-Net at 0.665. While BU²-Net and U-Net report similar IoU values, approximately 0.603 and 0.602 respectively, both substantially outperform U²-Net, which averages 0.571. For the loss metric, BU²-Net demonstrates superior optimization with a mean value of 1.128, notably lower than that of U²-Net at 1.298 and U-Net at 1.323. These comparisons are supported by very small p-values and large t-statistics, reinforcing the reliability of BU²-Net's convergence behavior.

Training dynamics further highlight BU²-Net's robustness. The dual-axis plots show a high degree of synchronization between the decrease in training loss and the corresponding improvement in recall, across six representative training runs. This pattern indicates a well-regulated optimization path. In contrast, U-Net exhibits unstable training behaviors, with early oscillations in loss and inconsistent recall trends. U²-Net's training is generally smoother but still suffers from divergence between loss and recall in some cases. BU²-Net maintains the most predictable learning trajectory, where optimization progress translates consistently into performance gains.

From a computational perspective, U²-Net achieved the shortest average epoch time and the lowest total training time per run, reflecting its lightweight design and fast convergence. U-Net, by contrast, had the longest training time and highest variability, indicating less stable optimization. BU²-Net fell between the two in terms of efficiency, incurring moderate additional computation due to the Bayesian components. However, this trade-off yielded significantly improved segmentation performance and training stability. These results suggest

that BU²-Net offers a favorable compromise between training efficiency and predictive accuracy, making it a practical option for tasks that demand both robustness and reliable performance.

Several limitations should be noted. First, although BU²-Net achieves strong performance in recall and loss, its IoU advantage over U-Net is marginal and statistically insignificant. This suggests that the observed improvement may be task-specific or sensitive to the characteristics of the dataset. Second, the model's architectural complexity, particularly the integration of Bayesian layers, may increase deployment overhead on resource-constrained platforms. This limitation highlights the need for further optimization to ensure efficient implementation in embedded applications. Third, although the model generally shows low variance, a few outlier runs still experience performance dips, indicating potential instability under broader initialization settings or data perturbations.

BU²-Net exhibits a favorable balance between predictive accuracy and computational cost. This trade-off is particularly relevant in small-sample learning scenarios, where computational resources are limited but accurate segmentation remains critical. Nonetheless, the current architecture still presents opportunities for further optimization, such as reducing parameter overhead or improving inference speed. To comprehensively assess its robustness and generalization, future evaluations on broader corrosion datasets under diverse lighting conditions are necessary.

6.3. Qualitative Analysis

This section builds on the quantitative findings by presenting a qualitative analysis of segmentation performance across a range of visually challenging scenarios. Representative samples from the test set illustrate the model’s ability to accurately localize corrosion regions

under conditions such as poor lighting, reflective metal surfaces, background clutter, and small-scale damage.

Activation map overlays and confidence-based visualizations are used to examine the spatial attention patterns and prediction behavior of the proposed Bayesian-U²-Net. These examples show how the model generalizes beyond ideal input conditions and responds to the inherent complexity of real-world corrosion imagery. The visual evidence complements statistical metrics by providing insights into boundary precision, error localization, and uncertainty modeling in practical deployment contexts.

6.3.1. Segmentation Comparison

A visual comparison of segmentation performance across six representative corrosion samples is presented in Fig. 24. Each row, indexed from (1) to (6) in bottom-to-top order, corresponds to a distinct testing scenario with varying levels of corrosion severity, background complexity, and lighting conditions. The columns display (a) the original RGB image, (b) the ground truth annotation, (c) the output of a traditional SVM-based method, (d) U-Net, (e) U²-Net, and (f) the proposed Bayesian U²-Net.

In the first two samples, all deep learning models outperform the SVM baseline in terms of region localization and segmentation quality. In sample (1), U-Net and U²-Net correctly identify the main corrosion area but introduce artifacts and irregular boundaries, while BU²-Net achieves a cleaner segmentation with smoother contours and fewer false positives. Sample (2), characterized by strong chromatic contrast and background noise, presents greater difficulty for all models. SVM shows many false detections, U-Net and U²-Net extract the key region with moderate accuracy, and BU²-Net demonstrates improved boundary precision and better shape restoration.

Sample (3) further illustrates the differences in model robustness under visually cluttered conditions. In this scene, which includes shadows and background pipes, both SVM and U-Net fail to detect the corrosion region. U²-Net captures part of the area, though with broken edges, whereas BU²-Net produces more consistent predictions with clearer structural continuity. In sample (4), where the corrosion signal is extremely faint, segmentation becomes more challenging for all models. SVM and U-Net entirely miss the corroded area, U²-Net recovers part of the region, and only BU²-Net generates a mask that closely aligns with the ground truth.

The final two samples reflect performance under occlusion, low contrast, and large-scale corrosion. In sample (5), SVM outputs are heavily impacted by lighting variation, while U-Net generates incomplete masks and U²-Net achieves partial reconstruction. BU²-Net maintains better regional completeness and robustness in the presence of visual degradation. Sample (6), involving a wide corroded surface, results in coarse and inaccurate boundaries for SVM, fragmented outputs from U-Net, and improved overall coverage by U²-Net. BU²-Net produces the most faithful and coherent segmentation, closely matching the ground truth. Together, these results suggest that BU²-Net consistently offers greater reliability in complex and degraded scenarios, while U²-Net and U-Net also provide substantial improvements over the traditional baseline.

The visual comparisons suggest that while U-Net and U²-Net improve upon traditional methods like SVM, each model has strengths and limitations depending on scene complexity. U²-Net benefits from multi-scale feature representation, and BU²-Net exhibits stronger robustness in preserving boundary details, reducing false detections, and maintaining segmentation continuity across challenging conditions. These findings highlight the importance of model selection based on application-specific requirements.

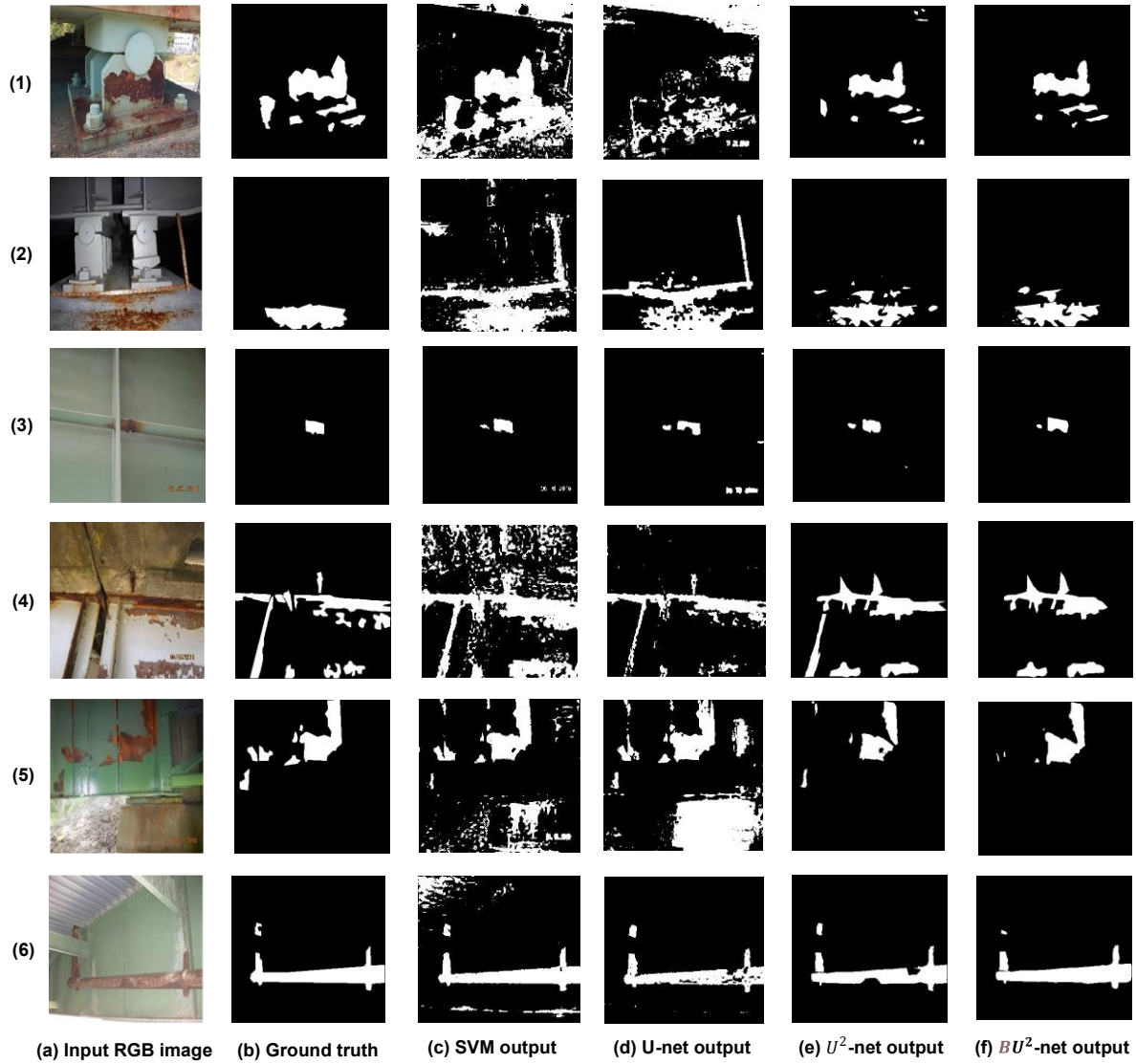


Fig. 24. Qualitative comparison of binary corrosion segmentation results.

6.3.2. Corrosion Activation Map

Activation map visualizations were employed to further examine the model's ability to localize corrosion regions. Representative samples from the test dataset are presented in Fig. 25, where each row contains the preprocessed input image and the corresponding corrosion activation map overlaid onto the original scene. All samples are sourced from the held-out test split of the publicly available COCO-Bridge dataset, ensuring reproducibility and fair evaluation.

The activation maps were extracted from the final segmentation outputs and visualized using a color heatmap, where warmer colors (e.g., red and yellow) indicate regions with higher

activation strength, suggesting a stronger network response to corrosion features. Cooler colors (e.g., blue and green) denote lower activation responses, typically corresponding to non-corroded background areas. By overlaying the activation maps onto the input images, the areas of focus are intuitively visualized, providing insight into the spatial attention mechanisms learned by the model.

In scenarios where strong surface reflections interfere with visual cues, accurate corrosion segmentation becomes particularly challenging. Reflective highlights on smooth metal surfaces or damp structural components may resemble corrosion patterns, increasing the risk of false positives. As illustrated by the heatmap results in Fig. 25, the model shows a degree of capability in mitigating such interference.

In the first and third columns, although pronounced specular reflections are present, the model largely maintains focused activations on the actual rusted areas without significant misclassification of adjacent glare. In the second and fourth columns, where reflections appear near bolts and seams that are commonly associated with corrosion, the model continues to exhibit reasonable corrosion sensitivity and spatial precision. These observations suggest that the model demonstrates a certain level of luminance invariance and contextual awareness, supporting relatively stable performance under challenging lighting conditions.

Fig. 25. Robust detection under low-light and reflective conditions. Left: preprocessed input; Right: activation map (warm = high response).

Fig. 26 illustrates the model's performance on large-scale corrosion regions. These scenarios often involve widespread rust spanning structural bases and connection joints, characterized by irregular shapes and complex material transitions. Such conditions can present challenges for accurate localization and boundary delineation.

The heatmaps indicate that the model generally activates across the primary corroded regions in most samples. In the first and second columns, although the textures are uneven and boundaries are not clearly defined, the model shows a tendency to focus on the main rust areas without generating excessive irrelevant activations. In the third and fourth columns, where corrosion is more fragmented and layered, the activations appear relatively continuous and aligned with the structural layout.

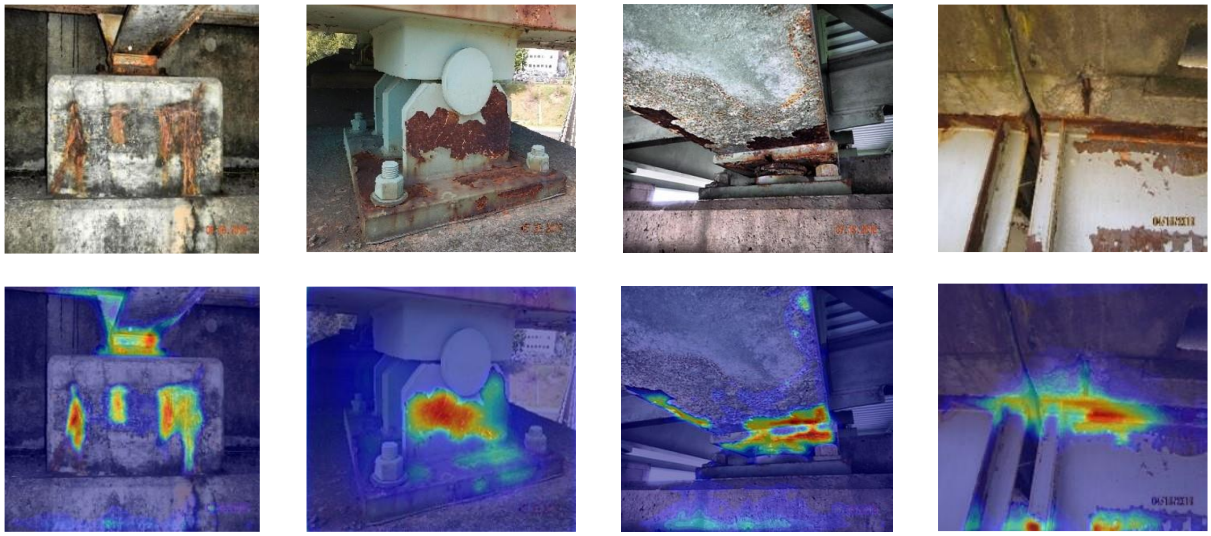


Fig. 26. Detection under background interference with similar patterns.

This set of visualizations highlights the model's response to fine-grained corrosion regions, particularly in cases where the affected areas are small, low in contrast, and easily overlooked. As shown in the heatmaps, the model appears to focus on subtle features along structural seams and edge joints, while generally avoiding activation in surrounding non-corroded regions. This tendency is observable in the first and third columns, where the corrosion is faint but still partially localized.

The second and fourth columns present more challenging scenarios involving clean surfaces, occlusion, and low-texture backgrounds. Even under these conditions, the model tends to maintain bounded activations and avoids major false positives. These preliminary observations, illustrated in Fig. 27, suggest that the model may have the potential to detect localized corrosion with reasonable sensitivity and spatial precision.

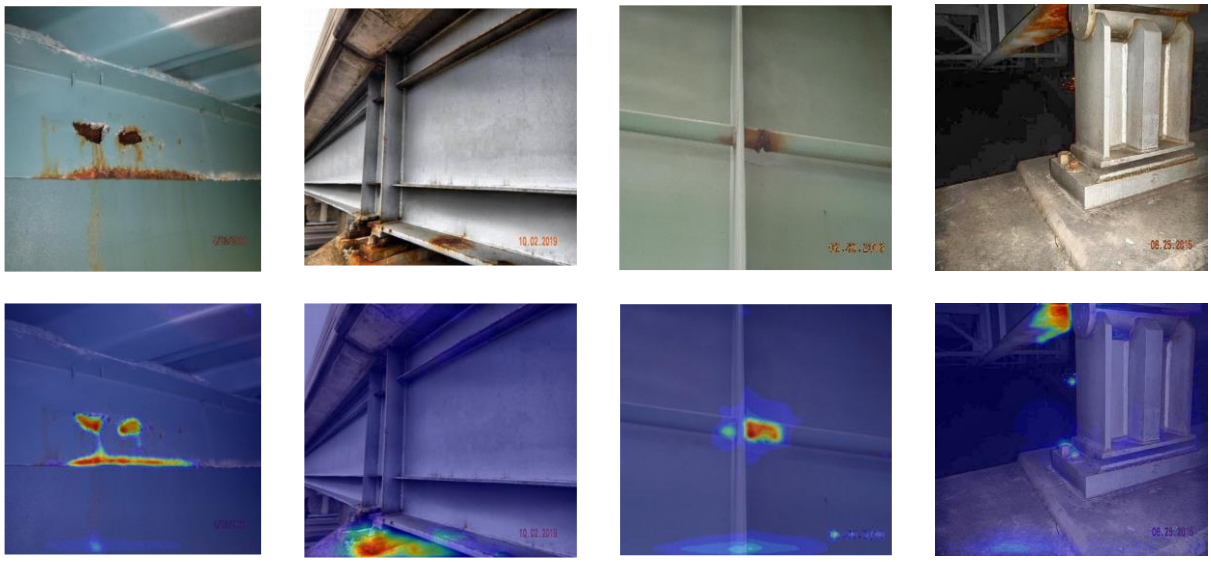


Fig. 27. Detection of localized and small corrosion areas.

In more challenging field scenarios, the model must cope not only with corrosion severity but also with complex structural geometry and background interference. The examples shown in Fig. 28 capture such cases, where the corrosion is extensive and often accompanied by overlapping metal layers, shadowed regions, and environmental clutter such as debris or staining. These conditions make boundary discrimination particularly difficult.

Despite these challenges, the activation maps indicate that the model tends to maintain spatial focus across corroded surfaces. In the first two columns, the model appears to localize corrosion extending across multiple interfaces, including vertical faces and layered concrete, while avoiding excessive activation in non-corroded areas. The third and fourth columns depict highly cluttered scenes with irregular textures and occlusions. In these cases, the model's attention remains relatively centered on areas likely associated with corrosion.

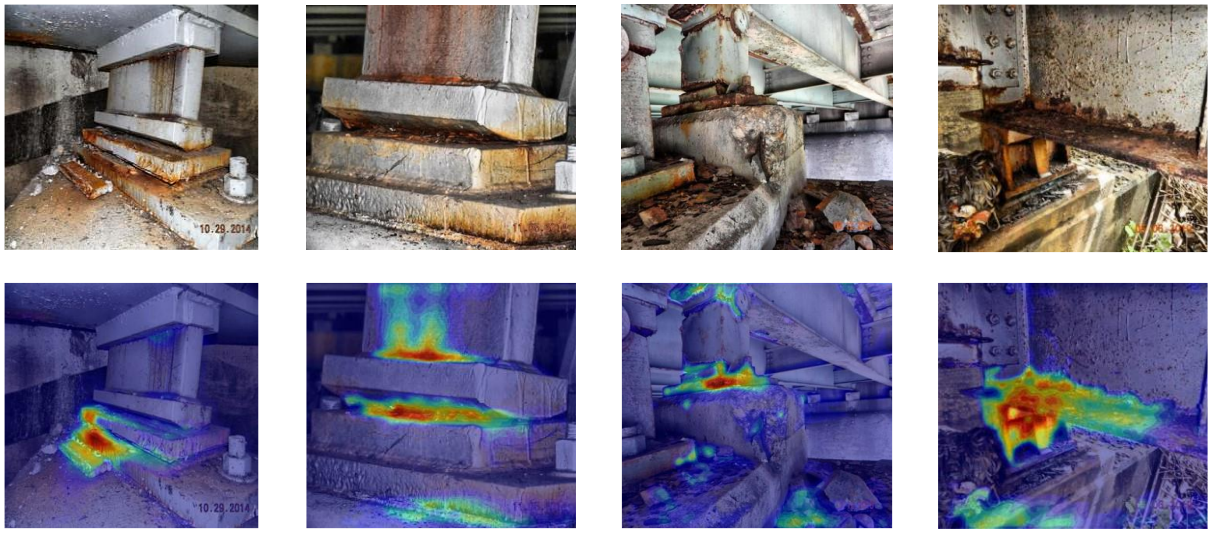


Fig. 28. Segmentation performance on structurally complex corrosion regions.

6.3.3. Confidence Map

To complement the quantitative evaluation, we conducted a visualization analysis to explore the interpretability of the Bayesian-U²-Net. Although the main purpose of using Bayesian convolutional layers via Bayes by Backprop (BBBConv) was to improve segmentation robustness and localization accuracy, the generated confidence and uncertainty maps also help reveal the model's behavior.

The confidence maps, shown in the first row of Fig. 29, were taken from the softmax output of the final segmentation layer. Bright areas represent a higher predicted probability of corrosion, mainly concentrated in the segmented rust regions, while darker areas correspond to either background or visually ambiguous regions. These maps provide a clear view of the model's prediction strength at the pixel level.

In parallel, epistemic uncertainty maps were generated using Monte Carlo sampling. During inference, the model was set to training mode to enable stochastic sampling in the BBBConv layers. Each image was passed through the network 2,000 times, and the pixel-wise variance of predictions was used to measure uncertainty. As shown in the second row of Fig. 29, regions

with high uncertainty often appear along corrosion boundaries, reflective surfaces, or ambiguous textures.

These visualizations reveal several patterns. In columns 1 and 3, the model shows high confidence in corrosion areas, while higher uncertainty appears near edges, reflecting sensitivity to spatial transitions. Column 2 shows high confidence but low uncertainty in a low-contrast region, suggesting possible overconfidence. In columns 4 and 5, where background textures are complex, the model maintains low uncertainty outside the corroded areas, indicating strong noise suppression. In column 5, moderate confidence with high uncertainty appears on metallic surfaces resembling rust, suggesting cautious behavior in uncertain cases.

Although full uncertainty modeling is not the focus of this work, these results show that lightweight Bayesian layers introduce meaningful prediction variance and support visual interpretability. However, the current setup does not separate epistemic from aleatoric uncertainty and may miss risks in low-texture or faint corrosion areas. Future work may consider richer variational methods or ensemble approaches to address these limitations.

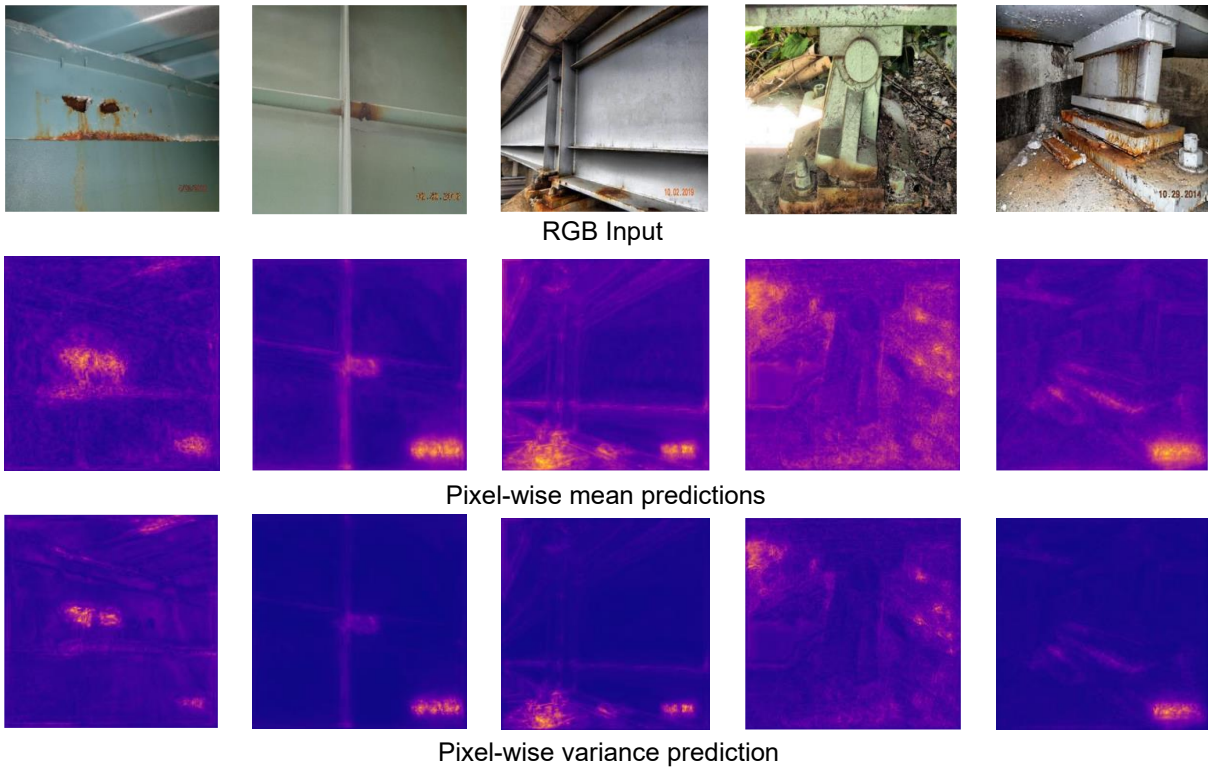


Fig. 29. Confidence and uncertainty maps from Bayesian-U²-Net. Top: sigmoid-based confidence; Bottom: variance-based uncertainty via Monte Carlo sampling.

6.3.4. Discussion

The qualitative results presented in this section suggest that the proposed Bayesian-U²-Net offers improved resilience in handling diverse visual disturbances. In scenarios with low illumination or surface reflections, the model shows a tendency to suppress irrelevant activations while maintaining reasonable focus on corrosion regions. It also exhibits the potential to better distinguish rust from visually similar background textures and to detect small-scale corrosion areas, though further validation on broader datasets would be beneficial to confirm these observations.

Closer inspection reveals several notable limitations. First, in highly ambiguous regions such as shadowed corners, corroded-metal interfaces, or rust-colored contaminants, the model occasionally suppresses activations too strongly, resulting in false negatives or partial omission of corrosion. This issue is especially visible in cases with low contrast or irregular corrosion shapes, where the activation maps appear fragmented or incomplete.

Second, although the integration of Bayesian convolutional layers improves feature robustness, it does not always lead to interpretable uncertainty behavior. Some visually ambiguous areas still exhibit low variance in the uncertainty maps, suggesting that the current approach may not fully capture the model's hesitation. This limitation may arise from relying solely on variational weight sampling, without incorporating aleatoric uncertainty or more expressive posterior approximations.

Third, the activation maps, although useful for visual interpretation, remain coarse at the boundaries and lack fine structural precision. In practical applications such as robotic inspection or corrosion quantification, this reduces their effectiveness unless combined with post-processing techniques.

The model shows strong qualitative generalization across varied corrosion conditions. At the same time, its limitations highlight the need for more adaptive attention mechanisms, multi-scale uncertainty modeling, and boundary-aware refinement modules. These findings also suggest that visual inspection alone may obscure subtle but systematic prediction biases, reinforcing the value of integrating qualitative interpretation with quantitative validation.

7. Chapter 7 –Conclusion

This study presents a novel deep learning framework for robust corrosion segmentation on steel structures, integrating illumination-adaptive preprocessing with GLF and LIME, and a Bayesian-enhanced U²-Net architecture. The proposed system addresses two key challenges in real-world UAV-based inspection: (i) performance degradation under uneven lighting and (ii) lack of uncertainty estimation in standard segmentation models.

The effectiveness of the proposed approach was validated on a UAV-acquired corrosion image dataset, using 45 independent training runs per model. Final performance was measured at the 100th training epoch on the validation set. Quantitative results show that BU²-Net outperforms both U²-Net and U-Net across all key metrics. Specifically:

- F1-score: BU²-Net achieves 75.088%, while U²-Net and U-Net achieve 72.437% and 69.590% respectively, representing relative improvements of +2.7 and +5.5 percentage points.
- IoU: BU²-Net scores 60.350%, slightly higher than U-Net at 60.195% and notably above U²-Net at 57.101%.
- Recall: BU²-Net attains 71.537%, outperforming U²-Net at 68.232% and U-Net at 66.456%, with relative gains of +3.3 and +5.1 percentage points.

In addition to higher accuracy, BU²-Net exhibits the lowest standard deviation across all metrics, indicating more stable and consistent performance across different runs. Compared to U²-Net, which displays wider variation and sensitivity to initialization, BU²-Net shows faster convergence, tighter training variance, and more compact output distributions, as confirmed by boxplots and training curve analyses.

The central innovation lies in the combined use of handcrafted illumination normalization and probabilistic deep learning. By introducing a fixed global-local enhancement pipeline prior to segmentation, the model benefits from improved visual consistency across heterogeneous UAV imagery. Meanwhile, the use of variational Bayesian layers within the U²-Net encoder introduces regularization and predictive uncertainty, enabling stronger generalization even under limited data conditions. Together, these contributions offer a practical and interpretable solution for corrosion segmentation that is both deployable on lightweight platforms and resilient in field conditions.

While the proposed framework demonstrates notable improvements in segmentation accuracy, stability, and real-world deployability, it remains limited to RGB imagery, which captures only surface-level visual features such as color, texture, and edges. This makes the model effective for routine inspections, but inadequate for capturing deeper structural manifestations such as early-stage subsurface degradation, hidden pitting, or microstructural fatigue. Consequently, the current approach cannot support mechanism-based diagnosis or long-term deterioration forecasting.

Corrosion is a complex multi-physics process involving electrochemical reactions, material fatigue, and environmental interactions across various spatial and temporal scales. The proposed segmentation model serves as a foundational step toward mechanism-aware understanding. Future work will integrate multimodal sensing data, including ultrasonic testing, scanning electron microscopy, and electrochemical impedance spectroscopy, with specialized deep learning models using mid-level attention and late-stage residual fusion. This approach aims to enable not only precise localization but also early warning and mechanistic insight, advancing structural health monitoring from visual recognition to predictive maintenance and full-lifecycle management.

8. Chapter 8 –Future work

8.1. Motivation

Recent advances in semantic segmentation have demonstrated the effectiveness of convolutional neural networks in detecting corrosion on steel surfaces. However, most existing models rely exclusively on RGB imagery, which captures only superficial visual characteristics such as color, texture, and shape. Although suitable for basic surface-level inspection, this modality is limited in addressing real-world complexities, including dirt contamination, coating deterioration, and variable lighting conditions. More critically, it cannot capture deeper structural phenomena such as corrosion-induced cracking or early-stage subsurface degradation. As a result, these approaches are inadequate for mechanism-aware diagnosis and long-term deterioration prediction.

Corrosion is not merely a surface anomaly but a multi-physics process involving electrochemical reactions, material fatigue, and environmental interaction. It exhibits distinct signatures across multiple spatial and temporal scales. For example, microstructural features such as grain boundary degradation and pitting cavities are best visualized using scanning electron microscopy or identified through X-ray diffraction. In contrast, ultrasonic testing and electrochemical impedance spectroscopy are essential for detecting hidden cracks, delamination, or early-stage subsurface transformation. These complexities demonstrate the need for a transition from vision-only approaches toward multi-modal fusion frameworks capable of integrating and interpreting diverse sensing data.

This study outlines a structured three-stage development pathway. The first stage, presented in Section 8.2, introduces a corrosion classification framework that connects visual severity-level annotation with mechanism-based typology. The proposed labeling system aligns with international standards as well as local infrastructure specifications. The second stage, detailed

in Section 8.3, designs a fusion-driven perception architecture. Each sensing modality, including SEM and RGB imagery, time-series signals, and electrochemical measurements, is paired with an appropriate deep learning model such as a Vision Transformer, LSTM network, or graph neural network. Fusion strategies are constructed using mid-level attention mechanisms and late-stage residual integration to achieve robust and interpretable predictions. The third stage, outlined in Section 8.4, proposes a cost-sensitive deployment roadmap based on implementation feasibility, training complexity, and hardware readiness. The resulting rollout plan prioritizes image and UT pipelines in the short term, promotes integrated fusion systems in the medium term, and anticipates full signal-based expansion incorporating EIS in longer-term research phases.

This research contributes to the evolution of corrosion diagnostics from surface-level observation to mechanism-informed understanding. The proposed framework supports intelligent structural health monitoring, the development of infrastructure-oriented digital twins, and predictive maintenance strategies. The capacity to sense, interpret, and respond to multi-scale corrosion behavior is essential for advancing the Construction 4.0 paradigm and ensuring the durability and resilience of critical infrastructure assets.

8.2. Corrosion Classification

8.2.1. Corrosion Severity

Initial efforts have been made toward establishing a corrosion classification framework. A three-tier severity labeling system was developed in accordance with the Waka Kotahi national standard for structural steel corrosion. To facilitate subsequent model training and evaluation, pixel-level annotations were completed across a dataset encompassing light, moderate, and severe corrosion cases. At present, the classification system operates on rule-based principles and has not yet been systematically integrated into model training pipelines. Future studies will investigate its performance within representative segmentation and detection architectures,

including instance segmentation using Mask R-CNN and object detection with YOLOv8, thereby enabling a comprehensive assessment of its adaptability and robustness. Furthermore, this labeling framework provides a foundation for lightweight deployment on edge devices, especially within UAV-assisted inspection scenarios and embedded corrosion monitoring applications.

Corrosion regions were classified into three severity levels based on the *Bridge Inspection and Maintenance Manual – Part 4: Structural Steel*, published by Waka Kotahi NZ Transport Agency. This national standard was selected for its alignment with current inspection and maintenance practices in New Zealand.

To improve global relevance and facilitate cross-study comparability, the classification scheme was cross-referenced with ISO 8501-1 (an international standard for surface preparation and coating condition) and the AASHTO Bridge Inspection Reference Manual (BIRM), commonly used for infrastructure inspection in the United States.

The resulting hybrid classification framework supports both regional applicability and international compatibility, making it suitable for a wide range of infrastructure scenarios. The three severity levels are defined as follows:

- **Class 1 (Minor):** Early-stage corrosion or surface discoloration, without significant metal loss.
- **Class 2 (Moderate):** Noticeable rusting and partial surface degradation, potentially indicating early structural concern.
- **Class 3 (Severe):** Extensive corrosion with evident metal loss, pitting, or structural damage.

Fig. 30 illustrates representative image samples along with their corresponding segmentation masks. The severity classes are visualized using green (Class 1), yellow (Class 2), and red (Class 3). Non-corrosive visual artifacts, including dirt, surface discoloration, and peeling paint, were intentionally excluded from the annotation process. This exclusion was implemented to prevent the incorporation of misleading visual cues during model training.

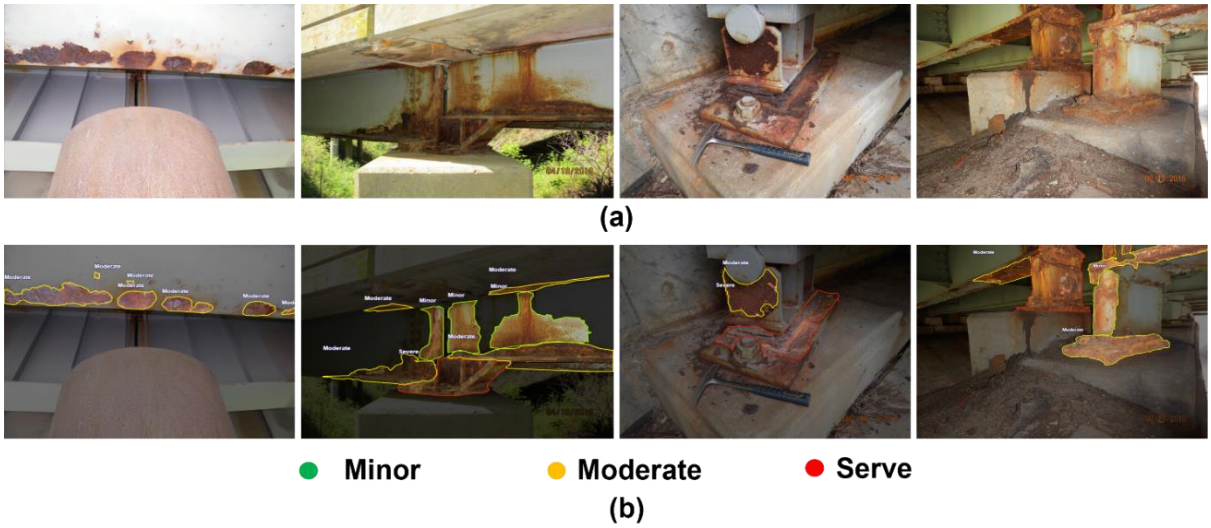


Fig. 30. (a) Sample images and (b) corresponding annotations

8.2.2. Corrosion Types

The current classification framework, despite its consistency with established inspection practices, remains fundamentally limited by the absence of a unified and fine-grained international standard for differentiating corrosion types. Existing guidelines, such as ISO 8501-1, depend primarily on visual approximation and manual interpretation, providing only a coarse assessment of severity without accounting for the underlying corrosion mechanisms. This lack of a standardized corrosion taxonomy poses a substantial obstacle to both model development and cross-study generalization.

Current classification systems do not distinguish among mechanisms such as pitting corrosion, crevice corrosion, galvanic corrosion, intergranular corrosion, or uniform surface corrosion.

Each of these forms varies in risk level, detectability, and impact on structural integrity, making mechanism-aware classification essential for prioritizing inspection and maintenance.

Current classification systems do not differentiate between corrosion mechanisms such as pitting corrosion, crevice corrosion, galvanic corrosion, intergranular corrosion, and uniform surface corrosion. These distinct forms differ significantly in terms of risk level, detectability, and their respective impacts on structural integrity. As a result, the development of mechanism-aware classification is essential for enabling more accurate prioritization of inspection and maintenance activities.

Table 8. Corrosion types considered for future mechanism-aware classification frameworks.

Corrosion Type	Cause	Environment	Chemical Reaction	Characteristics
1. Uniform	Surface oxidation	Air, water, acids	$Fe \rightarrow Fe^{2+} + 2e^{-} ½O_2 + H_2O + 2e^{-} \rightarrow 2OH^{-}$	Even thinning
2. Pitting	Film breakdown	Chloride, seawater	$Fe \rightarrow Fe^{2+} + 2e^{-}$ $O_2 + 4H^{+} + 4e^{-} \rightarrow 2H_2O$	Deep narrow pits
3. Crevice	Oxygen gradient	Gaskets, joints	$Fe \rightarrow Fe^{2+} + 2e^{-}$ $Fe^{2+} + 2H_2O \rightarrow Fe(OH)_2 + 2H^{+}$	Local acidification
4. Galvanic	Bimetallic coupling	Mixed metals in water	$Zn \rightarrow Zn^{2+} + 2e^{-}$ $O_2 + 2H_2O + 4e^{-} \rightarrow 4OH^{-}$	Fast anode loss
5. Intergranular	Grain boundary loss	Weld zones, stainless steel	$Fe \rightarrow Fe^{2+} + 2e^{-}$	Grain-edge attack
6. SCC	Stress + Cl^{-}	Tension + chlorides	$Fe \rightarrow Fe^{2+} + 2e^{-}$	Cracks under stress
7. Exfoliation	Subsurface swelling	Aluminum alloys	$Al \rightarrow Al^{3+} + 3e^{-} O_2 + 6e^{-} + 3H_2O \rightarrow 6OH^{-}$	Layered flaking

Corrosion Type	Cause	Environment	Chemical Reaction	Characteristics
8. Erosion	Flow + abrasion	High-velocity fluid	$Fe \rightarrow Fe^{2+} + 2e^{-}$	Grooves, material loss
9. MIC	Bacterial attack	Stagnant water, soil	$Fe + H_2S \rightarrow FeS + H_2\uparrow SO_4^{2-}$ + organic $\rightarrow H_2S$	Bio-pitting, black film

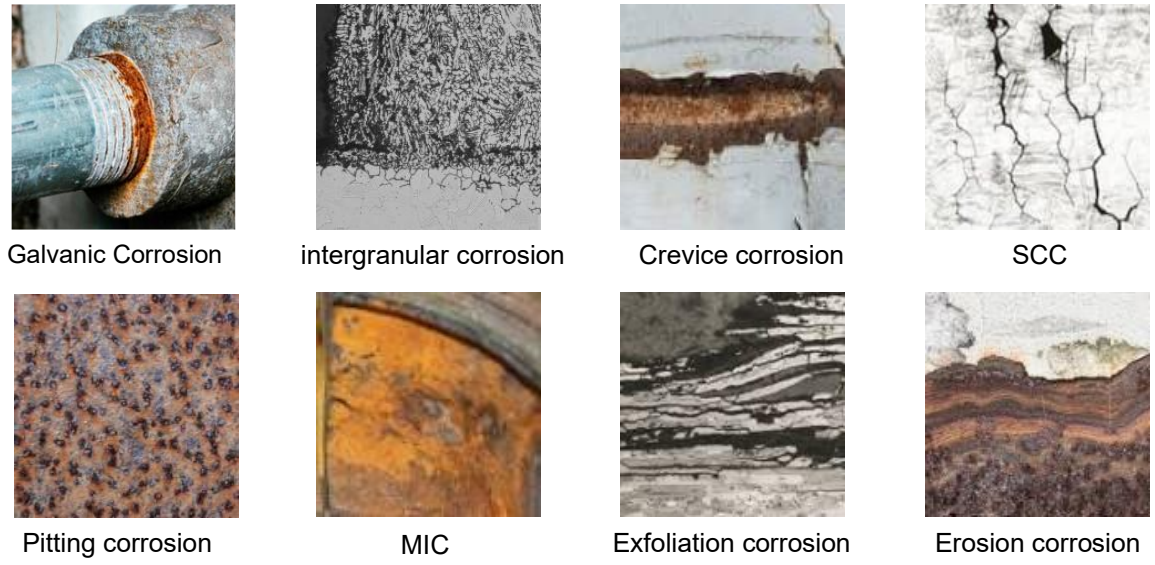


Fig. 31. Representative examples of different corrosion types.

The diverse physical and chemical characteristics of different corrosion types often render a single sensing modality, such as RGB imaging, inadequate for accurate classification or severity estimation. Several forms of corrosion, including pitting, crevice, and uniform surface corrosion, may appear visually similar while differing markedly in their underlying mechanisms, penetration depth, and implications for structural integrity. Consequently, selecting appropriate detection technologies tailored to specific corrosion types is essential for enabling mechanism-aware diagnosis.

Ultrasonic testing (UT), for example, is particularly effective in detecting intergranular corrosion and stress corrosion cracking (SCC), as it can reveal subsurface discontinuities and internal microcracks. Infrared thermography (IRT) is suitable for identifying crevice and pitting corrosion, especially when temperature gradients accentuate voids or oxidation zones. Electrochemical impedance spectroscopy (EIS) and open circuit potential (OCP) measurements provide real-time electrochemical feedback, allowing for differentiation between galvanic and uniform corrosion. In addition, microscopic techniques and X-ray diffraction (XRD) are instrumental in identifying microbiologically influenced corrosion (MIC) and dealloying phenomena such as dezincification, where surface chemistry plays a predominant role.

As infrastructure inspection shifts toward autonomous and robotic platforms, integrating these complementary sensing modalities through UAV-mounted payloads, embedded probes, or coordinated multi-sensor systems will be essential for achieving reliable and mechanism-sensitive corrosion monitoring under the Construction 4.0 paradigm.

Table 9. Suggested sensor modalities for mechanism-aware corrosion detection.

Corrosion Type	Recommended Detection Method
Uniform corrosion	RGB camera, thickness gauge
Pitting corrosion	Infrared thermography (IRT), ultrasonic
Crevice corrosion	IRT, moisture mapping sensors
Galvanic corrosion	Electrochemical sensors (OCP, EIS)
Intergranular corrosion	Ultrasonic testing, eddy current
Stress corrosion cracking	UT phased array, acoustic emission
Exfoliation corrosion	3D laser scanning, UT
Erosion corrosion	Visual + wall-thickness + flow monitoring
MIC	XRD, SEM, microbial DNA sampling

Traditional CNN-based segmentation methods are also restricted by the limitations of RGB imagery. Since RGB data captures only surface-level visual features, it often proves unreliable in conditions involving stains, aging coatings, or varying lighting environments. These constraints emphasize the inherent shortcomings of purely vision-based models. Progress in corrosion classification is therefore likely to depend on the integration of additional sensing modalities, such as ultrasonic data, infrared imaging, or surface conductivity measurements. These complementary data sources can be incorporated into intelligent robotic inspection platforms to improve diagnostic accuracy. Within this broader framework, the current task may be best understood as an efficient pre-screening process aimed at identifying regions of interest and expediting the overall inspection workflow. Future development will rely on the integration of machine perception with physical diagnostics to enable truly mechanism-aware corrosion monitoring systems.

8.3. Modal Fusion

8.3.1. Modal Matching

Within the broader landscape of corrosion mechanisms, only a subset poses direct and sustained threats to the mechanical integrity of steel infrastructure. From the perspective of both in-situ structural health monitoring and controlled laboratory evaluation, specific corrosion types must be prioritized due to their structural impact and detection complexity. Among these, pitting corrosion is especially critical, as it produces deep, narrow cavities that serve as stress risers, often initiating fatigue cracks. Stress corrosion cracking (SCC) is another high-risk mechanism, known to develop under tensile stress in the presence of aggressive ions, often leading to sudden and catastrophic failure even in the absence of significant material loss.

Uniform corrosion, while less abrupt in manifestation, is relevant due to its long-term cumulative effect: a gradual reduction in cross-sectional area and stiffness that undermines

structural load-bearing performance. In real-world bridge components and welded assemblies, galvanic corrosion frequently arises at dissimilar metal interfaces or electrically isolated joints, progressing rapidly under wet-dry cycling. In laboratory simulations, additional forms such as intergranular corrosion (common in stainless steel weld zones) and exfoliation corrosion (prevalent in layered aluminum alloys) are often included to replicate severe boundary-layer deterioration and stress-induced delamination.

These corrosion types span diverse physical mechanisms and spatial manifestations. Some are visibly distinct under RGB imaging (e.g., uniform rust films), while others, such as SCC or MIC-driven pitting, may exhibit minimal surface cues and require specialized modalities like ultrasonic pulse-echo, electrochemical impedance spectroscopy, or SEM/EDS surface characterization. Therefore, an effective structural corrosion monitoring framework must incorporate multi-modal sensing pathways tailored to the diagnostic requirements of these failure-relevant corrosion mechanisms. This necessity forms the technical basis for the integrated sensing architecture proposed in the subsequent sections.

Table 10. Alignment of major corrosion types

Corrosion Type	Structural Monitoring	Failure Testing	Typical Lab Tests
Pitting Corrosion	High-risk initiation sites, crack precursors	Crack initiation under stress	Tensile test, Fatigue test, Fracture toughness
SCC (Stress Corrosion Cracking)	Sudden failure, invisible surface cues	Critical for fracture mechanics study	SCC-specific slow strain rate test (SSRT), C-ring test
Uniform Corrosion	Long-term section loss monitoring	Low relevance to failure unless deep thinning	Corrosion rate test, Mass loss test
Galvanic Corrosion	Localized coupling zones, structural joints	Often used to simulate real joint behavior	Electrochemical cell tests, Galvanic series measurement

Corrosion Type	Structural Monitoring	Failure Testing	Typical Lab Tests
Intergranular Corrosion	Less visible in field, more microstructural	Important for weld zone analysis	ASTM A262 Practice E, Grain boundary etching
Exfoliation Corrosion	Layer delamination in aluminum components	Modeled in aging alloy structures	Bending fatigue, 3D profilometry, Cross-sectional SEM
MIC	Rarely evident in structure until advanced	Simulated in soil/water lab setups	Biofilm chamber, pH cycling, DNA sequencing on microbes

An effective multi-modal sensing architecture for corrosion diagnosis requires precise alignment between sensor modalities and computational models. Each sensing technique produces data with unique characteristics, including dense spatial textures from RGB and SEM imagery, structured time-series signals from ultrasonic testing (UT) and electrochemical impedance spectroscopy (EIS), and chemically specific spectral patterns from techniques such as X-ray diffraction (XRD) and X-ray fluorescence (XRF). As a result, different deep learning strategies must be employed to extract diagnostic features suited to the nature of each modality.

Two-dimensional imagery, such as RGB and SEM data, provides rich spatial information well-suited for convolutional neural networks (CNNs) and vision transformers. These models are proficient at recognizing surface textures, geometric patterns, and morphological defects. SEM imagery, typically collected under controlled laboratory conditions, is more effectively processed using classification-oriented architectures, such as ResNet or Vision Transformers (ViT), due to the dominance of localized microstructural features. In contrast, RGB imagery obtained from UAV-mounted or handheld systems is better aligned with pixel-wise segmentation models, such as U²-Net or DeepLab, which provide the spatial resolution necessary for in-situ corrosion mapping.

Signal-based modalities, including UT and EIS, require sequential modeling approaches capable of capturing time-dependent behaviors. Recurrent neural networks such as LSTM and GRU, as well as frequency-sensitive models like one-dimensional CNNs and transformers, can learn oscillatory trends, waveform distortions, and delay patterns that indicate the presence of subsurface flaws. When these signals originate from multiple spatially distributed points, graph neural networks (GNNs) offer an additional advantage by modeling structural relationships and signal correlations across components.

Based on the corrosion mechanisms prioritized in Section 8.3.1, including pitting corrosion, stress corrosion cracking (SCC), uniform corrosion, and galvanic coupling, the following sensor-model combinations are identified as particularly relevant:

- SEM imagery paired with ResNet or Vision Transformer for classification of corrosion morphology in laboratory conditions
- UT signals paired with LSTM or one-dimensional CNN for detecting the initiation and progression of subsurface cracks
- EIS data paired with Transformer or graph neural networks for classifying mechanism-specific electrochemical behavior

These core pairings form the architectural foundation of the multi-modal fusion framework discussed in the next section. By aligning each sensor's physical sensitivity with a computational model designed to exploit its strengths, the proposed system is expected to deliver both precise surface-level identification and robust mechanism-level inference.

8.3.2. Image Fusion

Designing an effective fusion strategy is essential for harnessing the complementary diagnostic capabilities of heterogeneous sensing modalities. Each sensor and model combination captures unique dimensions of corrosion behavior, including surface morphology, subsurface propagation, and electrochemical activity. Therefore, the fusion pipeline must be able to manage structural variability across different data types while maintaining the interpretability and task specificity of each individual signal pathway.

High-resolution scanning electron microscopy (SEM) offers essential visual access to microstructural corrosion patterns, capturing localized degradation phenomena such as pitting depth, grain boundary attack, and delamination in layered alloys. These texture-rich images exhibit high spatial frequency content and morphological discontinuities that provide strong indicators of the underlying corrosion mechanism. Given these characteristics, image classification backbones such as Residual Networks (ResNet) and Vision Transformers (ViT) are particularly effective at extracting discriminative features from SEM data.

An effective fusion framework involving SEM imagery can adopt two primary modes of integration. The first is intra-domain fusion, which consolidates SEM images acquired at different magnifications, spatial regions, or corrosion stages. This approach enables the model to learn scale-invariant representations and enhances its generalizability across varied acquisition conditions. The second is cross-domain fusion, in which SEM morphology images are aligned with elemental distribution maps obtained from techniques such as energy-dispersive X-ray spectroscopy (EDS) or X-ray diffraction (XRD). This alignment facilitates the joint modeling of geometric and compositional signatures of corrosion. In both fusion modes, SEM data serve as a visual proxy for localized material failure, while auxiliary modalities contribute to enhanced mechanistic interpretability.

Fusion strategies are commonly categorized as early or mid-level, based on the point of integration within the neural architecture. Early fusion involves stacking SEM morphology maps and corresponding chemical images into a multi-channel tensor, which is then processed by a shared convolutional backbone. This method leverages spatial alignment and shared filters to jointly model co-located texture and composition. In contrast, mid-level fusion is implemented through ViT-based encoders, where morphology and composition inputs are embedded through separate streams and later combined using cross-attention mechanisms. This configuration enables the network to associate visual structures, such as localized pitting, with material-specific features like the presence of Fe_2O_3 or chloride ions, thereby improving the mechanistic accuracy of classification outcomes.

This morphology-driven fusion approach is particularly well-suited for laboratory-based corrosion studies, where imaging conditions are controlled and annotations are highly reliable. The combination of SEM data with ViT-based models supports advanced downstream tasks, including failure origin attribution, corrosion stage prediction, and the generation of synthetic data for domain adaptation. Moreover, integrating cross-domain channels such as EDS improves the interpretability of the classification process, aligns with materials science principles, and fosters trust in intelligent diagnostic systems. Future research may investigate multi-view SEM fusion, attention-guided region-of-interest segmentation, or integration with graph-based encoders capable of modeling morphological transitions across spatial patches.

8.3.3. Signal Modeling

Ultrasonic Testing (UT) remains a foundational technique in non-destructive evaluation (NDE), offering rich time-domain signals that reveal internal material defects. These signals, which include echo delay patterns, amplitude attenuation, and multi-interface reflections, contain critical information regarding the initiation, trajectory, and propagation of subsurface cracks.

Unlike image-based modalities, UT produces sequential one-dimensional (1D) waveform data, which necessitates specialized temporal modeling techniques for effective diagnostic interpretation.

Two complementary modeling strategies are particularly well suited to capturing the temporal dependencies within ultrasonic signals: Long Short-Term Memory (LSTM) networks and one-dimensional Convolutional Neural Networks (1D-CNNs). LSTM models are effective in representing the progression of echoes over time, especially when addressing pulse-sequence variability and subtle delay shifts. In parallel, 1D-CNNs can identify localized frequency-domain characteristics such as high-frequency attenuation and waveform distortion, making them suitable for low-level feature extraction prior to sequential modeling.

A hierarchical modeling framework is proposed to combine these approaches:

- Low-level signal encoding involves applying sliding-window 1D-CNN filters to raw UT waveforms in order to extract local spatiotemporal features across the frequency and amplitude domains.
- High-level sequence modeling uses LSTM encoders to process the extracted feature sequences, enabling the model to learn patterns associated with degradation processes such as fatigue crack development and delamination.

For integration into a multi-modal system, UT-derived features can be fused with spatial modalities such as RGB or infrared imagery through late fusion techniques. In this configuration, cross-modal attention modules may be employed to allow ultrasonic features to influence visual attention maps, particularly in cases where surface appearance remains unchanged despite internal damage. This approach proves especially useful in detecting fatigue-induced microcracks in structural joints, where visual modalities alone are often insufficient for early-stage detection.

Further research may explore the development of spatial ultrasonic graphs, wherein distributed UT sensor arrays installed across structural components are represented as nodes. By encoding both geometric adjacency and signal correlation into the graph structure, Graph Neural Networks (GNNs) can be utilized to jointly model spatially distributed crack propagation and inter-component degradation dynamics. Such graph-based modeling approaches hold significant promise for real-time structural health monitoring in complex environments characterized by heterogeneous stress distributions.

8.3.4. EIS Fusion

Electrochemical Impedance Spectroscopy (EIS) provides a powerful modality for analyzing underlying corrosion mechanisms by measuring the frequency-dependent response of metal-electrolyte interfaces. Through impedance measurements across a broad frequency spectrum, typically visualized using Nyquist or Bode plots, EIS captures essential electrochemical phenomena such as charge-transfer resistance, double-layer capacitance, diffusion impedance, and polarization behavior. These features vary significantly depending on corrosion type, environmental exposure, and material composition, making EIS an indispensable tool for mechanism-aware diagnostics.

Despite its diagnostic utility, interpreting EIS data presents significant challenges. The data are high-dimensional and spectrally nonlinear, and impedance behavior is influenced by both global corrosion progression and localized electrochemical activity. Extracting meaningful patterns and classifying corrosion types requires deep learning models that can capture long-range dependencies and complex multivariate relationships.

Transformer-based architectures have recently gained attention as a suitable solution for modeling EIS data. Their self-attention mechanisms allow the network to learn intricate correlations between impedance values across different frequencies. This enables the model to

focus on important spectral regions, such as high-frequency semicircles that reflect coating degradation or low-frequency Warburg tails that indicate diffusion limitations. In contrast to recurrent neural networks, Transformers process the entire frequency spectrum simultaneously, preserving global structure and improving robustness in the presence of noise or sparse sampling.

When EIS data are obtained from spatially distributed sensor arrays, such as multi-electrode configurations on structural surfaces, each time-resolved impedance profile can be represented as a node in a graph. Graph Neural Networks (GNNs) are well suited for modeling the spatial and electrochemical relationships between sensor locations. In this representation, edge weights may encode physical distance, material similarity, or correlated degradation patterns, while node features consist of local impedance vectors and additional metadata, including temperature, pH, or chloride concentration.

A representative fusion architecture may include the following components:

- Frequency-domain encoding: Raw impedance vectors, including real and imaginary components, are passed through a Transformer encoder to produce a global spectral representation that captures the nonlinear impedance behavior associated with specific corrosion mechanisms.
- Spatial correlation modeling: These encoded features are assigned to graph nodes, and a GNN propagates contextual information across the network, enabling the identification of spatially linked degradation processes such as galvanic corrosion or multi-site microbial activity.
- Mid-level multi-modal fusion: When EIS data are co-registered with imaging modalities such as RGB or SEM, the electrochemical features can be concatenated or cross-attended with visual representations extracted by CNN or ViT models. This

allows the system to jointly reason about chemical activity and surface degradation patterns, such as connecting high polarization resistance with early-stage rust formation.

This integrated framework is particularly effective for distinguishing between corrosion types that exhibit similar visual appearances but different electrochemical characteristics. For example, microbiologically influenced corrosion (MIC) often results in localized pitting accompanied by sharp low-frequency impedance shifts and changes in charge-transfer resistance due to biofilm formation. These features may be difficult to detect using visual data alone.

To enhance predictive accuracy, multi-resolution fusion techniques can be incorporated. These approaches aggregate spectral features from different frequency ranges into a hierarchical structure, enabling the model to recognize both coarse and fine-scale patterns. Additionally, temporal graph attention networks may be used to track impedance changes over time, supporting the forecasting of corrosion progression across inspection cycles.

The integration of Transformer and GNN-based encoders creates a flexible and mechanism-aware corrosion analysis system that can interpret both localized electrochemical behavior and broader spatial interactions. This capability is crucial for deploying intelligent inspection platforms that support early risk identification, corrosion mechanism discrimination, and proactive maintenance planning in complex infrastructure environments.

The integration of three fusion pathways, including scanning electron microscopy with Vision Transformers for microstructural classification, ultrasonic testing with LSTM networks for temporal crack modeling, and electrochemical impedance spectroscopy with Transformer and graph neural network architectures for electrochemical mechanism identification, establishes a cohesive and scalable framework for intelligent corrosion monitoring. Each pathway is precisely matched to the data structure and diagnostic strengths of its respective sensing

modality, enabling the system to capture visual, physical, and chemical characteristics of corrosion across spatial and temporal dimensions.

Future implementations may adopt a modular architecture in which these branches operate independently while converging at a unified decision layer. This structure supports integrated predictions that are interpretable and mechanism-aware. It also improves generalization under real-world variability and facilitates the deployment of robust multi-sensor corrosion analysis in both laboratory and field environments.

8.4. Fusion Planning

In parallel, the integration of ultrasonic testing (UT) into corrosion detection pipelines offers a practical approach for subsurface flaw identification. Unlike surface imaging techniques, UT provides access to internal discontinuities and early-stage defect initiation, which are essential for identifying conditions such as stress corrosion cracking or subsurface fatigue. By coupling these time-series signals with vision-based surface features through mid-level or late-stage fusion architectures, the system can extend its diagnostic capabilities without requiring expensive imaging hardware upgrades. More broadly, this integration strategy aligns with established practices in structural health monitoring and enables dual-function utility in infrastructure asset management beyond corrosion-specific applications.

A rational investment strategy for corrosion sensing must consider not only technical novelty but also time-to-deployment, resource availability, and system integration complexity. The comparative cost-benefit of each modality and fusion path should be evaluated across five key dimensions:

8.4.1. Time & Cycle

The integration of scanning electron microscopy with convolutional neural networks offers the shortest research and development cycle. This is primarily due to the widespread availability

of microscopy equipment and pre-trained deep learning models. The acquisition of morphological data, image annotation, and model convergence can typically be completed within a few weeks, making this approach highly suitable for pilot studies and early-stage academic prototyping.

In comparison, the combination of ultrasonic testing with LSTM or one-dimensional CNN architectures involves moderate complexity, particularly in signal preprocessing and feature extraction. However, this pathway benefits from a well-established body of literature in structural health monitoring and can be implemented within a medium-term development timeline.

The pathway involving electrochemical impedance spectroscopy combined with Transformer or graph-based architectures represents a longer-term investment. EIS systems require complex setup, calibration, and expert interpretation. Signal modeling is further challenged by spectral noise, class imbalance, and the need for reliable ground truth generation. As such, this modality is best suited to institutions with access to advanced electrochemical laboratories and long-duration research funding.

Conclusion: SEM and UT-based systems support short to medium-term development. EIS integration is more appropriate for long-term strategic research.

8.4.2. Resource Need

SEM-CNN pipelines are computationally efficient, particularly when leveraging transfer learning. However, they require significant manual effort for sample preparation, micrograph annotation, and corrosion morphology interpretation. Once operational, the model training and inference process can be conducted on standard GPU-based platforms.

UT-based models using LSTM or one-dimensional CNN architectures demand signal processing expertise, but the computational requirements are relatively low. These models can

run on edge devices such as Jetson Nano or Raspberry Pi 4, enabling real-time deployment in the field.

In contrast, EIS pipelines built on Transformer or GNN models present the highest barrier to entry. They require interdisciplinary expertise in electrochemistry, circuit modeling, and spectral signal analysis. Additionally, the computational cost of Transformer training and GNN message passing is high, often necessitating access to dedicated computing servers.

Conclusion: SEM and UT approaches are resource-efficient and accessible for most research teams. EIS integration demands both specialized personnel and computational infrastructure.

8.4.3. Access

SEM data is widely available in academic institutions and materials laboratories. The acquisition process is standardized, with established imaging protocols and labeling workflows.

UT equipment is moderately accessible, with a range of low-cost probes and portable devices available commercially. However, signal quality is sensitive to surface conditions and material interfaces, requiring noise mitigation and calibration protocols.

In contrast, EIS setups are relatively expensive and require environmental shielding. Public datasets are limited and often focused on chemical corrosion studies rather than structural applications. As a result, custom data collection is frequently necessary, which increases the time and cost associated with adoption.

Conclusion: SEM offers the highest level of accessibility. UT provides a good balance between availability and field relevance. EIS remains constrained to specialized laboratory environments.

8.4.4. Deployment

SEM-based CNN models are effective for high-resolution post-inspection analysis but are generally unsuitable for real-time or embedded applications due to the constraints of imaging hardware.

UT-based models demonstrate strong deployment potential. Their low memory requirements and compatibility with embedded processors make them suitable for real-world implementation. These models have already been validated in UAV-based inspection and portable structural health monitoring platforms.

EIS-based models remain limited in deployment maturity. The computational demands, latency, and sensitivity to environmental factors hinder their current feasibility for embedded or distributed use. While promising for mechanism attribution, such systems are best suited for centralized diagnostics.

Conclusion: UT offers the most deployable architecture for field applications. SEM is best suited for offline diagnostics. EIS remains more appropriate for controlled environments.

8.4.5. Fusion Maturity

Fusion involving SEM and RGB imagery is the most technically mature. The data types share dimensional compatibility, and common model backbones can be used. Both early fusion via input stacking and mid-level fusion via attention mechanisms in Vision Transformers are well documented and widely implemented.

Fusion between UT and visual modalities such as RGB is moderately mature. Synchronization between modalities remains challenging, but late fusion methods such as dual-branch networks and attention-based fusion layers have demonstrated good performance in recent studies.

Cross-domain fusion involving EIS is still in its early stages. Bridging electrochemical signals with visual or structural data requires co-calibrated datasets and joint physical modeling frameworks, which are not yet commonly available in the literature.

Conclusion: RGB-SEM fusion is currently the most integration-ready. UT-based fusion continues to improve in maturity. EIS fusion still requires foundational advances in representation alignment and multi-domain learning.

8.4.6. Roadmap

Based on the above analysis, a tiered deployment plan is proposed:

Table 11. Deployment plan

Phase	Priority Modality	Strategy Description
Short-Term	SEM + CNN	Morphological classification via ViT/ResNet in lab conditions
Mid-Term	UT + LSTM/1D-CNN	Subsurface defect detection using edge-deployable time-series models
Long-Term	EIS + Transformer/GNN	Electrochemical mechanism identification via advanced fusion

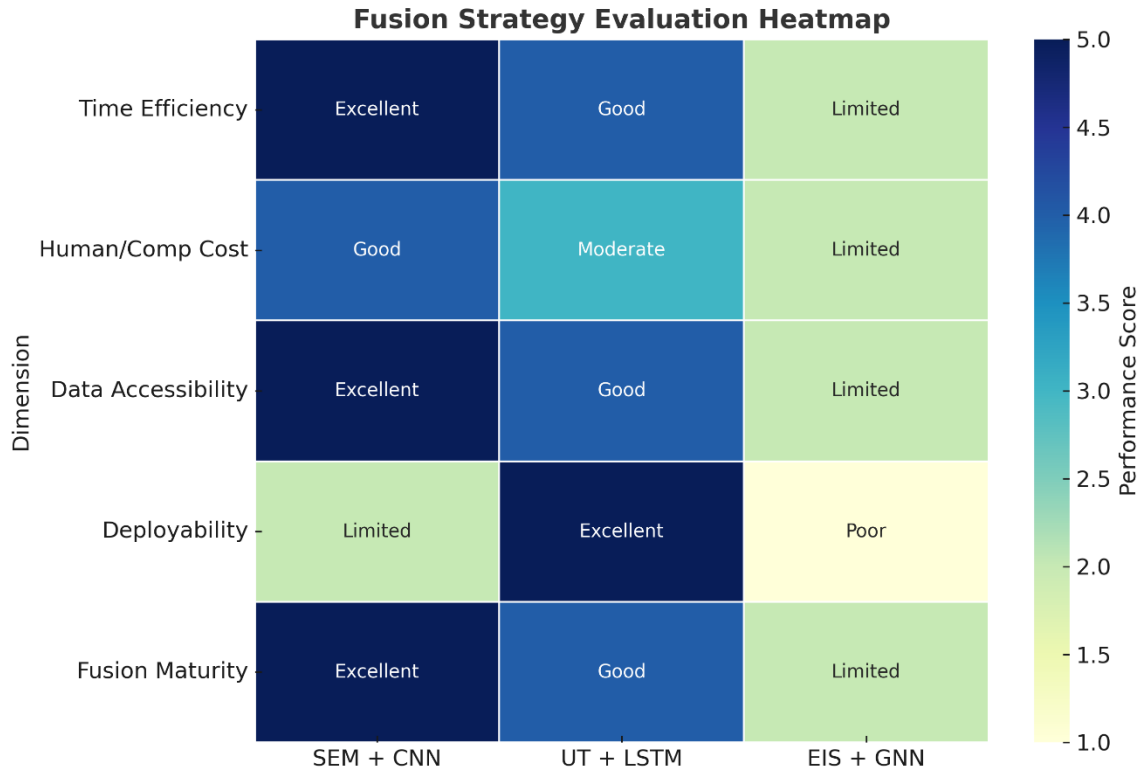


Fig. 32. Evaluation Heatmap

These modalities are not mutually exclusive. The SEM and UT pathways can be co-developed in the short term with unified output formats and synchronized input interfaces, establishing a modular base for eventual multi-modal integration that includes EIS. This staged investment strategy maximizes early deliverables while enabling scalable expansion to complex, mechanism-aware diagnostics in future research cycles.

References

[1] Z. C. Petrović, ‘Catastrophes caused by corrosion’, *Vojnoteh. Glas.*, vol. 64, no. 4, pp. 1048–1064, 2016, doi: 10.5937/vojtehg64-10388.

[2] G. Koch, ‘Cost of corrosion’, in *Trends in Oil and Gas Corrosion Research and Technologies*, Elsevier, 2017, pp. 3–30. doi: 10.1016/B978-0-08-101105-8.00001-2.

[3] A. Ortiz, F. Bonnin-Pascual, E. Garcia-Fidalgo, and J. P. Company, ‘Visual Inspection of Vessels by Means of a Micro-Aerial Vehicle: An Artificial Neural Network Approach for Corrosion Detection’, in *Robot 2015: Second Iberian Robotics Conference*, L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano, and V. Muñoz-Martinez, Eds., Cham: Springer International Publishing, 2016, pp. 223–234. doi: 10.1007/978-3-319-27149-1_18.

[4] I. Katsamenis, N. Doulamis, A. Doulamis, E. Protopapadakis, and A. Voulodimos, ‘Simultaneous Precise Localization And Classification of metal rust defects for robotic-driven maintenance and prefabrication using residual attention U-Net’, *Autom. Constr.*, vol. 137, p. 104182, May 2022, doi: 10.1016/j.autcon.2022.104182.

[5] E. Forcael, I. Ferrari, A. Opazo-Vega, and J. A. Pulido-Arcas, ‘Construction 4.0: A Literature Review’, *Sustainability*, vol. 12, no. 22, Art. no. 22, Jan. 2020, doi: 10.3390/su12229755.

[6] M. Khayatazad, L. De Pue, and W. De Waele, ‘Detection of corrosion on steel structures using automated image processing’, *Dev. Built Environ.*, vol. 3, p. 100022, Aug. 2020, doi: 10.1016/j.dibe.2020.100022.

[7] R. Vorobel, I. Ivasenko, O. Berehulyak, and T. Mandzii, ‘Segmentation of rust defects on painted steel surfaces by intelligent image analysis’, *Autom. Constr.*, vol. 123, p. 103515, Mar. 2021, doi: 10.1016/j.autcon.2020.103515.

[8] K.-W. Liao and Y.-T. Lee, ‘Detection of rust defects on steel bridge coatings via digital image recognition’, *Autom. Constr.*, vol. 71, pp. 294–306, Nov. 2016, doi: 10.1016/j.autcon.2016.08.008.

[9] S. Livens *et al.*, ‘A Texture Analysis Approach to Corrosion Image Classification’, *Microsc. Microanal. Microstruct.*, vol. 7, no. 2, pp. 143–152, Apr. 1996, doi: 10.1051/mmm:1996110.

[10] ‘Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection’. Accessed: Apr. 19, 2025. [Online]. Available:

https://journals.sagepub.com/doi/epdf/10.1177/1475921717737051?src=getftr&utm_source=sciencedirect_contenthosting&getft_integrator=sciencedirect_contenthosting

[11] E. Protopapadakis, I. Katsamenis, and A. Doulamis, 'Multi-label deep learning models for continuous monitoring of road infrastructures', in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, in PETRA '20. New York, NY, USA: Association for Computing Machinery, Jun. 2020, pp. 1–7. doi: 10.1145/3389189.3397997.

[12] X. Zhang *et al.*, 'A Deep Learning-Based Approach for Automated Yellow Rust Disease Detection from High-Resolution Hyperspectral UAV Images', *Remote Sens.*, vol. 11, no. 13, Art. no. 13, Jan. 2019, doi: 10.3390/rs11131554.

[13] Y.-J. Cha, W. Choi, and O. Büyüköztürk, 'Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks', *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, 2017, doi: 10.1111/mice.12263.

[14] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, 'A deep learning-based framework for an automated defect detection system for sewer pipes', *Autom. Constr.*, vol. 109, p. 102967, Jan. 2020, doi: 10.1016/j.autcon.2019.102967.

[15] I. Katsamenis, E. Protopapadakis, A. Voulodimos, D. Dres, and D. Drakoulis, 'Man overboard event detection from RGB and thermal imagery: possibilities and limitations', in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, in PETRA '20. New York, NY, USA: Association for Computing Machinery, Jun. 2020, pp. 1–6. doi: 10.1145/3389189.3397998.

[16] I. Katsamenis, E. Protopapadakis, A. Doulamis, N. Doulamis, and A. Voulodimos, 'Pixel-Level Corrosion Detection on Metal Constructions by Fusion of Deep Learning Semantic and Contour Segmentation', in *Advances in Visual Computing*, G. Bebis, Z. Yin, E. Kim, J. Bender, K. Subr, B. C. Kwon, J. Zhao, D. Kalkofen, and G. Baciu, Eds., Cham: Springer International Publishing, 2020, pp. 160–169. doi: 10.1007/978-3-030-64556-4_13.

[17] I. Katsamenis, E. Protopapadakis, A. Doulamis, N. Doulamis, and A. Voulodimos, 'Pixel-Level Corrosion Detection on Metal Constructions by Fusion of Deep Learning Semantic and Contour Segmentation', in *Advances in Visual Computing*, G. Bebis, Z. Yin, E. Kim, J. Bender, K. Subr, B. C. Kwon, J. Zhao, D. Kalkofen, and G. Baciu, Eds., Cham: Springer International Publishing, 2020, pp. 160–169. doi: 10.1007/978-3-030-64556-4_13.

[18] C. Kasperzyk, M.-K. Kim, and I. Brilakis, ‘Automated re-prefabrication system for buildings using robotics’, *Autom. Constr.*, vol. 83, pp. 184–195, Nov. 2017, doi: 10.1016/j.autcon.2017.08.002.

[19] L. Jaillon, C. S. Poon, and Y. H. Chiang, ‘Quantifying the waste reduction potential of using prefabrication in building construction in Hong Kong’, *Waste Manag.*, vol. 29, no. 1, pp. 309–320, Jan. 2009, doi: 10.1016/j.wasman.2008.02.015.

[20] C. M. Tam, ‘- Impact on structure of labour market resulting from large-scale implementation of prefabrication’, in *Advances in Building Technology*, M. Anson, J. M. Ko, and E. S. S. Lam, Eds., Oxford: Elsevier, 2002, pp. 399–403. doi: 10.1016/B978-008044100-9/50052-8.

[21] A. Khan, S. S. A. Ali, A. Anwer, S. H. Adil, and F. Mériaudeau, ‘Subsea Pipeline Corrosion Estimation by Restoring and Enhancing Degraded Underwater Images’, *IEEE Access*, vol. 6, pp. 40585–40601, 2018, doi: 10.1109/ACCESS.2018.2855725.

[22] W. Nash, L. Zheng, and N. Birbilis, ‘Deep learning corrosion detection with confidence’, *Npj Mater. Degrad.*, vol. 6, no. 1, pp. 1–13, Mar. 2022, doi: 10.1038/s41529-022-00232-6.

[23] C. Chen, Q. Chen, J. Xu, and V. Koltun, ‘Learning to See in the Dark’, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 3291–3300. doi: 10.1109/CVPR.2018.00347.

[24] A. M. Reza, ‘Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement’, *J. VLSI Signal Process. Syst. Signal Image Video Technol.*, vol. 38, no. 1, pp. 35–44, Aug. 2004, doi: 10.1023/B:VLSI.0000028532.53893.82.

[25] J. Bai, Y. Yin, Q. He, Y. Li, and X. Zhang, ‘Retinexmamba: Retinex-based Mamba for Low-light Image Enhancement’, May 20, 2024, *arXiv*: arXiv:2405.03349. doi: 10.48550/arXiv.2405.03349.

[26] Q. Hu *et al.*, ‘Inter-Rater Uncertainty Quantification in Medical Image Segmentation via Rater-Specific Bayesian Neural Networks’, Aug. 25, 2023, *arXiv*: arXiv:2306.16556. doi: 10.48550/arXiv.2306.16556.

[27] H. Xiong, S. Liu, R. V. Sharan, E. Coiera, and S. Berkovsky, ‘Weak label based Bayesian U-Net for optic disc segmentation in fundus images’, *Artif. Intell. Med.*, vol. 126, p. 102261, Apr. 2022, doi: 10.1016/j.artmed.2022.102261.

[28] L. Mou, L. Liang, Z. Gao, and X. Wang, ‘A multi-scale anomaly detection framework for retinal OCT images based on the Bayesian neural network’, *Biomed. Signal Process. Control*, vol. 75, p. 103619, May 2022, doi: 10.1016/j.bspc.2022.103619.

[29] Q.-C. Tian and L. D. Cohen, ‘A variational-based fusion model for non-uniform illumination image enhancement via contrast optimization and color correction’, *Signal Process.*, vol. 153, pp. 210–220, Dec. 2018, doi: 10.1016/j.sigpro.2018.07.022.

[30] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, ‘U2-Net: Going deeper with nested U-structure for salient object detection’, *Pattern Recognit.*, vol. 106, p. 107404, Oct. 2020, doi: 10.1016/j.patcog.2020.107404.

[31] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, ‘Weight Uncertainty in Neural Network’, in *International Conference on Machine Learning*, PMLR, Jun. 2015, pp. 1613–1622. Accessed: Apr. 21, 2025. [Online]. Available: https://proceedings.mlr.press/v37/blundell15.html

[32] P.-H. Chen and L.-M. Chang, ‘Artificial intelligence application to bridge painting assessment’, *Autom. Constr.*, vol. 12, no. 4, pp. 431–445, Jul. 2003, doi: 10.1016/S0926-5805(03)00016-5.

[33] S. Lee, L.-M. Chang, and P.-H. Chen, ‘Performance Comparison of Bridge Coating Defect Recognition Methods’, *Corrosion*, vol. 61, no. 1, pp. 12–20, Jan. 2005, doi: 10.5006/1.3278155.

[34] M. Enikeev, I. Gubaydullin, and M. Maleeva, ‘Analysis of Corrosion Process Development on Metals by Means of Computer Vision’, *Eng. J.*, vol. 21, no. 4, pp. 183–192, Jul. 2017, doi: 10.4186/ej.2017.21.4.183.

[35] S. Lee, L.-M. Chang, and M. Skibniewski, ‘Automated recognition of surface defects using digital color image processing’, *Autom. Constr.*, vol. 15, no. 4, pp. 540–549, Jul. 2006, doi: 10.1016/j.autcon.2005.08.001.

[36] P.-H. Chen, Y.-C. Chang, L.-M. Chang, and P. C. Doerschuk, ‘Application of Multiresolution Pattern Classification to Steel Bridge Coating Assessment’, *J. Comput. Civ. Eng.*, vol. 16, no. 4, pp. 244–251, Oct. 2002, doi: 10.1061/(ASCE)0887-3801(2002)16:4(244).

[37] R. M. Pidaparti, B. S. Aghazadeh, A. Whitfield, A. S. Rao, and G. P. Mercier, ‘Classification of corrosion defects in NiAl bronze through image analysis’, *Corros. Sci.*, vol. 52, no. 11, pp. 3661–3666, Nov. 2010, doi: 10.1016/j.corsci.2010.07.017.

[38] I. Ivasenko and V. Chervatyuk, ‘Detection of Rust Defects of Protective Coatings Based on HSV Color Model’, in *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, Jul. 2019, pp. 1143–1146. doi: 10.1109/UKRCON.2019.8879804.

[39] P.-H. Chen, Y.-C. Yang, and L.-M. Chang, ‘Automated bridge coating defect recognition using adaptive ellipse approach’, *Autom. Constr.*, vol. 18, no. 5, pp. 632–643, Aug. 2009, doi: 10.1016/j.autcon.2008.12.007.

[40] T. Gibbons, G. Pierce, K. Worden, and I. Antoniadou, ‘A Gaussian Mixture Model for Automated Corrosion Detection in Remanufacturing’, in *Advances in Manufacturing Technology XXXII*, IOS Press, 2018, pp. 63–68. doi: 10.3233/978-1-61499-902-7-63.

[41] L. Petricca, T. Moss, G. Figueroa, and S. Broen, ‘Corrosion Detection Using A.I : A Comparison of Standard Computer Vision Techniques and Deep Learning Model’, in *Computer Science & Information Technology (CS & IT)*, Academy & Industry Research Collaboration Center (AIRCC), May 2016, pp. 91–99. doi: 10.5121/csit.2016.60608.

[42] K. Y. Choi and S. S. Kim, ‘Morphological analysis and classification of types of surface corrosion damage by digital image processing’, *Corros. Sci.*, vol. 47, no. 1, pp. 1–15, Jan. 2005, doi: 10.1016/j.corsci.2004.05.007.

[43] F. N. S. Medeiros, G. L. B. Ramalho, M. P. Bento, and L. C. L. Medeiros, ‘On the Evaluation of Texture and Color Features for Nondestructive Corrosion Detection’, *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 1, p. 817473, Dec. 2010, doi: 10.1155/2010/817473.

[44] M. Aliofkhazraei, *Developments in Corrosion Protection*. BoD – Books on Demand, 2014.

[45] M. P. Bento and G. L. B. Ramalho, ‘Image Processing Techniques applied for Corrosion Damage Analysis’.

[46] F. Bonnin-Pascual and A. Ortiz, ‘Corrosion Detection for Automated Visual Inspection’, in *Developments in Corrosion Protection*, M. Aliofkhazraei, Ed., InTech, 2014. doi: 10.5772/57209.

[47] P.-H. Chen, H.-K. Shen, C.-Y. Lei, and L.-M. Chang, ‘Support-vector-machine-based method for automated steel bridge rust assessment’, *Autom. Constr.*, vol. 23, pp. 9–19, May 2012, doi: 10.1016/j.autcon.2011.12.001.

[48] F. Tsutsumi, H. Murata, T. Onoda, O. Oguri, and H. Tanaka, ‘Automatic corrosion estimation using galvanized steel images on power transmission towers’, in *2009 Transmission & Distribution Conference & Exposition: Asia and Pacific*, Oct. 2009, pp. 1–4. doi: 10.1109/TD-ASIA.2009.5356961.

[49] H.-K. Shen, P.-H. Chen, and L.-M. Chang, ‘Automated steel bridge coating rust defect recognition method based on color and texture feature’, *Autom. Constr.*, vol. 31, pp. 338–356, May 2013, doi: 10.1016/j.autcon.2012.11.003.

[50] B. Valeti and S. Pakzad, ‘Automated Detection of Corrosion Damage in Power Transmission Lattice Towers Using Image Processing’, pp. 474–482, Apr. 2017, doi: 10.1061/9780784480427.040.

[51] X. Quintana, E. Martinez, and J. Melenchon, ‘Automatic evaluation of degradation of paint coatings through EM algorithm’, in *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, Sep. 2003, p. III–565. doi: 10.1109/ICIP.2003.1247307.

[52] H.-K. Shen, P.-H. Chen, and L.-M. Chang, ‘Human-visual-perception-like intensity recognition for color rust images based on artificial neural network’, *Autom. Constr.*, vol. 90, pp. 178–187, Jun. 2018, doi: 10.1016/j.autcon.2018.02.023.

[53] H. Son, N. Hwang, C. Kim, and C. Kim, ‘Rapid and automated determination of rusted surface areas of a steel bridge for robotic maintenance systems’, *Autom. Constr.*, vol. 42, pp. 13–24, Jun. 2014, doi: 10.1016/j.autcon.2014.02.016.

[54] S. Mabbutt, P. Picton, P. Shaw, and S. Black, ‘Review of Artificial Neural Networks (ANN) applied to corrosion monitoring’, *J. Phys. Conf. Ser.*, vol. 364, no. 1, p. 012114, May 2012, doi: 10.1088/1742-6596/364/1/012114.

[55] H. Furuta, T. Deguchi, and M. Kushida, ‘Neural network analysis of structural damage due to corrosion’, in *Proceedings of 3rd International Symposium on Uncertainty Modeling and Analysis and Annual Conference of the North American Fuzzy Information Processing Society*, Sep. 1995, pp. 109–114. doi: 10.1109/ISUMA.1995.527678.

[56] S. Livens *et al.*, ‘Classification of corrosion images by wavelet signatures and LVQ networks’, in *Computer Analysis of Images and Patterns*, V. Hlaváč and R. Šára, Eds., Berlin, Heidelberg: Springer, 1995, pp. 538–543. doi: 10.1007/3-540-60268-2_341.

[57] M. R. Jahanshahi and S. F. Masri, ‘Effect of Color Space, Color Channels, and Sub-Image Block Size on the Performance of Wavelet-Based Texture Analysis Algorithms: An

Application to Corrosion Detection on Steel Structures', pp. 685–692, Aug. 2013, doi: 10.1061/9780784413029.086.

[58] D. L. Naik, H. U. Sajid, R. Kiran, and G. Chen, 'Detection of Corrosion-Indicating Oxidation Product Colors in Steel Bridges under Varying Illuminations, Shadows, and Wetting Conditions', *Metals*, vol. 10, no. 11, Art. no. 11, Nov. 2020, doi: 10.3390/met10111439.

[59] T. E. Potok *et al.*, 'A Study of Complex Deep Learning Networks on High-Performance, Neuromorphic, and Quantum Computers', *J Emerg Technol Comput Syst*, vol. 14, no. 2, p. 19:1-19:21, Jul. 2018, doi: 10.1145/3178454.

[60] T. Young, D. Hazarika, S. Poria, and E. Cambria, 'Recent Trends in Deep Learning Based Natural Language Processing [Review Article]', *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018, doi: 10.1109/MCI.2018.2840738.

[61] H.-K. Shen, P.-H. Chen, and L.-M. Chang, 'Human-visual-perception-like intensity recognition for color rust images based on artificial neural network', *Autom. Constr.*, vol. 90, pp. 178–187, Jun. 2018, doi: 10.1016/j.autcon.2018.02.023.

[62] G. Yao, T. Lei, and J. Zhong, 'A review of Convolutional-Neural-Network-based action recognition', *Pattern Recognit. Lett.*, vol. 118, pp. 14–22, Feb. 2019, doi: 10.1016/j.patrec.2018.05.018.

[63] A. Dhillon and G. K. Verma, 'Convolutional neural network: a review of models, methodologies and applications to object detection', *Prog. Artif. Intell.*, vol. 9, no. 2, pp. 85–112, Jun. 2020, doi: 10.1007/s13748-019-00203-0.

[64] D. J. Atha and M. R. Jahanshahi, 'Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection', *Struct. Health Monit.*, vol. 17, no. 5, pp. 1110–1128, Sep. 2018, doi: 10.1177/1475921717737051.

[65] Y. Ma, Y. Yang, Y. Yao, S. Li, and X. Zhao, 'Image-based corrosion recognition for ship steel structures', in *Smart Structures and NDE for Industry 4.0*, SPIE, Mar. 2018, pp. 143–149. doi: 10.1117/12.2296540.

[66] J. Du, L. Yan, H. Wang, and Q. Huang, 'Research on grounding grid corrosion classification method based on convolutional neural network', *MATEC Web Conf.*, vol. 160, p. 01008, 2018, doi: 10.1051/matecconf/201816001008.

[67] J. Feng, F. Li, S. Lu, J. Liu, and D. Ma, 'Injurious or Noninjurious Defect Identification From MFL Images in Pipeline Inspection Using Convolutional Neural Network', *IEEE*

Trans. Instrum. Meas., vol. 66, no. 7, pp. 1883–1892, Jul. 2017, doi: 10.1109/TIM.2017.2673024.

[68] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, 'Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types', *Comput.-Aided Civ. Infrastruct. Eng.*, vol. 33, no. 9, pp. 731–747, 2018, doi: 10.1111/mice.12334.

[69] D. Kang, 'Autonomous unmanned aerial vehicles and deep learning-based damage detection', Nov. 2021, Accessed: Apr. 23, 2025. [Online]. Available: http://hdl.handle.net/1993/36120

[70] R. Girshick, J. Donahue, T. Darrell, and J. Malik, 'Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation'.

[71] 'Girshick_Fast_R-CNN_ICCV_2015_paper'.

[72] S. Ren, K. He, R. Girshick, and J. Sun, 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[73] A. Rahman, Z. Y. Wu, and R. Kalfarisi, 'Semantic Deep Learning Integrated with RGB Feature-Based Rule Optimization for Facility Surface Corrosion Detection and Evaluation', *J. Comput. Civ. Eng.*, vol. 35, no. 6, p. 04021018, Nov. 2021, doi: 10.1061/(ASCE)CP.1943-5487.0000982.

[74] J. Long, E. Shelhamer, and T. Darrell, 'Fully Convolutional Networks for Semantic Segmentation'.

[75] W. Liu, A. Rabinovich, and A. C. Berg, 'ParseNet: Looking Wider to See Better', Nov. 19, 2015, *arXiv*: arXiv:1506.04579. doi: 10.48550/arXiv.1506.04579.

[76] H. Li, P. Xiong, J. An, and L. Wang, 'Pyramid Attention Network for Semantic Segmentation', Nov. 25, 2018, *arXiv*: arXiv:1805.10180. doi: 10.48550/arXiv.1805.10180.

[77] X. Li *et al.*, 'Weighted Feature Pyramid Networks for Object Detection', in *2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2019, pp. 1500–1504. doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217.

[78] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, ‘Pyramid Scene Parsing Network’, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI: IEEE, Jul. 2017, pp. 6230–6239. doi: 10.1109/CVPR.2017.660.

[79] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, ‘DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[80] W. Nash, T. Drummond, and N. Birbilis, ‘Quantity beats quality for semantic segmentation of corrosion in images’, Jun. 30, 2018, *arXiv*: arXiv:1807.03138. doi: 10.48550/arXiv.1807.03138.

[81] V. Hoskere, Y. Narazaki, T. Hoang, and B. S. Jr, ‘Vision-based Structural Inspection using Multiscale Deep Convolutional Neural Networks’, May 02, 2018, *arXiv*: arXiv:1805.01055. doi: 10.48550/arXiv.1805.01055.

[82] T. Tong, J. Lin, J. Hua, F. Gao, and H. Zhang, ‘Crack identification for bridge condition monitoring using deep convolutional networks trained with a feedback-update strategy’, *Maint. Reliab. Cond. Monit.*, vol. 1, no. 2, Art. no. 2, Dec. 2021, doi: 10.21595/mrcm.2021.22032.

[83] O. Ronneberger, P. Fischer, and T. Brox, ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[84] T. Nguyen *et al.*, ‘U-Net for MAV-based Penstock Inspection: an Investigation of Focal Loss in Multi-class Segmentation for Corrosion Identification’, Sep. 18, 2018, *arXiv*: arXiv:1809.06576. doi: 10.48550/arXiv.1809.06576.

[85] J. Shi *et al.*, ‘Improvement of Damage Segmentation Based on Pixel-Level Data Balance Using VGG-Unet’, *Appl. Sci.*, vol. 11, no. 2, Art. no. 2, Jan. 2021, doi: 10.3390/app11020518.

[86] A. Kendall and Y. Gal, ‘What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?’, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Apr. 24, 2025. [Online]. Available:

https://proceedings.neurips.cc/paper_files/paper/2017/hash/2650d6089a6d640c5e85b2b88265dc2b-Abstract.html

[87] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, 'Natural Adversarial Examples', in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 15257–15266. doi: 10.1109/CVPR46437.2021.01501.

[88] M. Khan, D. Nielsen, V. Tangkaratt, W. Lin, Y. Gal, and A. Srivastava, 'Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam', in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Jul. 2018, pp. 2611–2620. Accessed: Apr. 24, 2025. [Online]. Available: https://proceedings.mlr.press/v80/khan18a.html

[89] K. Osawa *et al.*, 'Practical Deep Learning with Bayesian Principles', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019. Accessed: Apr. 24, 2025. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/hash/b53477c2821c1bf0da5d40e57b870d35-Abstract.html

[90] F. K. Gustafsson, M. Danelljan, and T. B. Schon, 'Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision', in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 1289–1298. doi: 10.1109/CVPRW50498.2020.00167.

[91] A. Kendall, V. Badrinarayanan, and R. Cipolla, 'Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding', Oct. 10, 2016, *arXiv*: arXiv:1511.02680. doi: 10.48550/arXiv.1511.02680.

[92] K. Shridhar, F. Laumann, and M. Liwicki, 'A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference', Jan. 08, 2019, *arXiv*: arXiv:1901.02731. doi: 10.48550/arXiv.1901.02731.

[93] A. G. Wilson, 'The Case for Bayesian Deep Learning', Jan. 29, 2020, *arXiv*: arXiv:2001.10995. doi: 10.48550/arXiv.2001.10995.

[94] X. Chen, E.-J. Park, and D. Xiu, 'A flexible numerical approach for quantification of epistemic uncertainty', *J. Comput. Phys.*, vol. 240, pp. 211–224, May 2013, doi: 10.1016/j.jcp.2013.01.018.

[95] J. Jakeman, M. Eldred, and D. Xiu, ‘Numerical approach for quantification of epistemic uncertainty’, *J. Comput. Phys.*, vol. 229, no. 12, pp. 4648–4663, Jun. 2010, doi: 10.1016/j.jcp.2010.03.003.

[96] K. Shridhar, F. Laumann, and M. Liwicki, ‘A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference’, Jan. 08, 2019, *arXiv*: arXiv:1901.02731. doi: 10.48550/arXiv.1901.02731.

[97] A. G. Wilson, ‘The Case for Bayesian Deep Learning’, Jan. 29, 2020, *arXiv*: arXiv:2001.10995. doi: 10.48550/arXiv.2001.10995.

[98] E. H. Land and J. J. McCann, ‘Lightness and Retinex Theory’, *JOSA*, vol. 61, no. 1, pp. 1–11, Jan. 1971, doi: 10.1364/JOSA.61.000001.

[99] J. J. McCann, ‘Retinex at 50: color theory and spatial algorithms, a review’, *J. Electron. Imaging*, vol. 26, no. 3, p. 031204, Feb. 2017, doi: 10.1117/1.JEI.26.3.031204.

[100] J. J. McCann and A. Rizzi, *The Art and Science of HDR Imaging*. John Wiley & Sons, 2011.

[101] D. Zosso, G. Tran, and S. J. Osher, ‘Non-Local Retinex---A Unifying Framework and Beyond’, *SIAM J. Imaging Sci.*, vol. 8, no. 2, pp. 787–826, Jan. 2015, doi: 10.1137/140972664.

[102] G. Stockman and L. G. Shapiro, *Computer Vision*, 1st ed. USA: Prentice Hall PTR, 2001.

[103] J. A. Frankle and J. J. McCann, ‘Method and apparatus for lightness imaging’, US4384336A, May 17, 1983 Accessed: Apr. 24, 2025. [Online]. Available: https://patents.google.com/patent/US4384336A/en

[104] D. J. Jobson, Z. Rahman, and G. A. Woodell, ‘Properties and performance of a center/surround retinex’, *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 451–462, Mar. 1997, doi: 10.1109/83.557356.

[105] D. J. Jobson, Z. Rahman, and G. A. Woodell, ‘A multiscale retinex for bridging the gap between color images and the human observation of scenes’, *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 965–976, Jul. 1997, doi: 10.1109/83.597272.

[106] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel, ‘A Variational Framework for Retinex’, *Int. J. Comput. Vis.*, vol. 52, no. 1, pp. 7–23, Apr. 2003, doi: 10.1023/A:1022314423998.

[107] E. H. Land, ‘An alternative technique for the computation of the designator in the retinex theory of color vision.’, *Proc. Natl. Acad. Sci.*, vol. 83, no. 10, pp. 3078–3080, May 1986, doi: 10.1073/pnas.83.10.3078.

[108] R. Sharma and A. Kamra, ‘A Review on CLAHE Based Enhancement Techniques’, in *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, Sep. 2023, pp. 321–325. doi: 10.1109/IC3I59117.2023.10397722.

[109] P. K. Verma, N. P. Singh, and D. Yadav, ‘Image Enhancement: A Review’, in *Ambient Communications and Computer Systems*, Y.-C. Hu, S. Tiwari, M. C. Trivedi, and K. K. Mishra, Eds., Singapore: Springer, 2020, pp. 347–355. doi: 10.1007/978-981-15-1518-7_29.

[110] E. Bianchi, A. L. Abbott, P. Tokekar, and M. Hebdon, ‘COCO-Bridge: Structural Detail Data Set for Bridge Inspections’, *J. Comput. Civ. Eng.*, vol. 35, no. 3, p. 04021003, May 2021, doi: 10.1061/(ASCE)CP.1943-5487.0000949.

[111] L. A. Szolga, ‘On Flight Real Time Image Processing by Drone Equipped with Raspberry Pi4’, in *2021 IEEE 27th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, Oct. 2021, pp. 334–337. doi: 10.1109/SIITME53254.2021.9663650.

[112] L. Zhou *et al.*, ‘UAV vision-based crack quantification and visualization of bridges: system design and engineering application’, *Struct. Health Monit.*, vol. 24, no. 2, pp. 1083–1100, Mar. 2025, doi: 10.1177/14759217241251778.

[113] F. Wang *et al.*, ‘Rapid in-flight image quality check for UAV-enabled bridge inspection’, *ISPRS J. Photogramm. Remote Sens.*, vol. 212, pp. 230–250, Jun. 2024, doi: 10.1016/j.isprsjprs.2024.05.008.

[114] J. H. Lee, S. Yoon, B. Kim, G.-H. Gwon, I.-H. Kim, and H.-J. Jung, ‘A new image-quality evaluating and enhancing methodology for bridge inspection using an unmanned aerial vehicle’, *Smart Struct. Syst.*, vol. 27, no. 2, pp. 209–226, 2021, doi: 10.12989/SSS.2021.27.2.209.

[115] ‘Roboflow Universe: Computer Vision Datasets’, Roboflow. Accessed: Apr. 05, 2025. [Online]. Available: https://universe.roboflow.com/

[116] G.-H. Gwon, J. H. Lee, I.-H. Kim, and H.-J. Jung, ‘CNN-Based Image Quality Classification Considering Quality Degradation in Bridge Inspection Using an Unmanned

Aerial Vehicle', *IEEE Access*, vol. 11, pp. 22096–22113, 2023, doi: 10.1109/ACCESS.2023.3238204.

[117] T.-Y. Lin *et al.*, 'Microsoft COCO: Common Objects in Context', in *Computer Vision – ECCV 2014*, vol. 8693, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., in Lecture Notes in Computer Science, vol. 8693. , Cham: Springer International Publishing, 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.

[118] J. Pineau, P. Vincent-Lamarre, K. Sinha, V. Lariviere, and A. Beygelzimer, 'Improving Reproducibility in Machine Learning Research'.

[119] C. Shorten and T. M. Khoshgoftaar, 'A survey on Image Data Augmentation for Deep Learning', *J. Big Data*, vol. 6, no. 1, p. 60, Jul. 2019, doi: 10.1186/s40537-019-0197-0.

[120] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, 'Albumentations: Fast and Flexible Image Augmentations', *Information*, vol. 11, no. 2, Art. no. 2, Feb. 2020, doi: 10.3390/info11020125.

[121] L. Zhao, Y. Wang, F. Bu, P. Wang, L. Tian, and C. Liu, 'Corrosion damage detection and evaluation of coated steel components under multiple illumination conditions', *Measurement*, vol. 250, p. 117179, Jun. 2025, doi: 10.1016/j.measurement.2025.117179.

[122] D. L. Naik, H. U. Sajid, R. Kiran, and G. Chen, 'Detection of Corrosion-Indicating Oxidation Product Colors in Steel Bridges under Varying Illuminations, Shadows, and Wetting Conditions', *Metals*, vol. 10, no. 11, Art. no. 11, Nov. 2020, doi: 10.3390/met10111439.

[123] Z. Ameli, S. J. Nesheli, and E. N. Landis, 'Deep Learning-Based Steel Bridge Corrosion Segmentation and Condition Rating Using Mask RCNN and YOLOv8', *Infrastructures*, vol. 9, no. 1, p. 3, Dec. 2023, doi: 10.3390/infrastructures9010003.

[124] D. J. Jobson, Z. Rahman, and G. A. Woodell, 'A multiscale retinex for bridging the gap between color images and the human observation of scenes', *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 965–976, Jul. 1997, doi: 10.1109/83.597272.

[125] Z.-U. Rahman, G. A. Woodell, and D. J. Jobson, 'A Comparison of the Multiscale Retinex With Other Image Enhancement Techniques'. Jan. 01, 1997. Accessed: Jun. 08, 2025. [Online]. Available: https://ntrs.nasa.gov/citations/20040110657

[126] X. Guo, Y. Li, and H. Ling, 'LIME: Low-Light Image Enhancement via Illumination Map Estimation', *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 982–993, Feb. 2017, doi: 10.1109/TIP.2016.2639450.

[127] W. Wang, X. Wu, X. Yuan, and Z. Gao, ‘An Experiment-Based Review of Low-Light Image Enhancement Methods’, *IEEE Access*, vol. 8, pp. 87884–87917, 2020, doi: 10.1109/ACCESS.2020.2992749.

[128] B. Yin, ‘Corrosion Image Data Set for Automating Scientific Assessment of Materials’.

[129] O. Ronneberger, P. Fischer, and T. Brox, ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[130] C. Cortes and V. Vapnik, ‘Support-vector networks’, *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/BF00994018.

[131] O. Ronneberger, P. Fischer, and T. Brox, ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.