Heritage Defect Detection Under Data Scarcity: A Leakage-Aware YOLO–Faster R-CNN Ensemble with Weighted Boxes Fusion

by

Lamia Yusuf
21201098
Faiaz Zahin
21101090
Sumaiya Sultana
21201223
Mohammad Ishmam Bin Aslam
21201041
Ahmed Tahlil Kadir
21101138

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

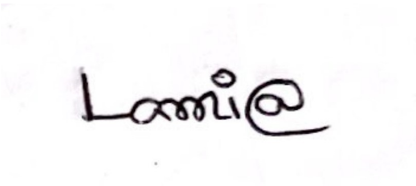
Department of Computer Science and Engineering
Brac University
October, 2025.

Declaration

It is hereby declared that

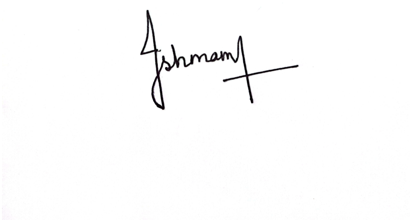
1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Lamia Yusuf
21201098

Faiaz Zahin
21101090



Sumaiya Sultana
21201223

Mohammad Ishmam Bin Aslam
21201041

Ahmed Tahil Kadir
21101138

Approval

The thesis titled "Heritage Defect Detection Under Data Scarcity: A Leakage-Aware YOLO–Faster R-CNN Ensemble with Weighted Boxes Fusion " submitted by

1. Lamia Yusuf (21201098)
2. Faiaz Zahin(21101090)
3. Sumaiya Sultana (21201223)
4. Mohammad Ishmam Bin Aslam (21201041)
5. Ahmed Tahlil Kadir (21101138)

of Summer 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in 2025 Year.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Dibyo Fabian Dofadar
Lecturer
Department of Computer Science and Engineering
BRAC University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Enginnering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Environmental, biological, and structural reasons are causing cultural heritage sites in Bangladesh to deteriorate more quickly. At the same time, traditional manual inspection procedures are still time-consuming, subjective, and hard to scale. This study introduces an automated deep learning framework for the detection and classification of surface flaws in historical structures, utilizing the Historic Place Dataset including 2,292 photos from Panam City, a UNESCO World Heritage Site in Sonargaon, Bangladesh. We use Weighted Boxes Fusion (WBF) to combine YOLOv8-small for speed and Faster R-CNN for accuracy to find five types of damage: Artistic, Corroded brick, Corroded plaster, Fungus, and Living plant elements. To address the class imbalance and limited data, we use hybrid augmentation methodologies that use both online and offline methods. The experimental results demonstrate that Faster R-CNN has the best test-set performance (mAP@0.5 = 0.8945), and the WBF ensemble (mAP@0.5 = 0.8898) does much better than YOLOv8 alone (mAP@0.5 = 0.864). Per-class analysis shows that features that are easy to tell apart get better results, but damage types that are hard to tell apart are still hard to work with. This study shows that deep learning methods can work even when there isn't a lot of data available, and it also shows how to improve them through systematic augmentation and architectural optimization.

Keywords: Heritage defect detection, Deep learning, YOLOv8, Faster R-CNN, Weighted Boxes Fusion, Object detection, Cultural heritage preservation, Bangladesh, Class imbalance, Data augmentation, Ensemble learning, Automated damage assessment, Computer vision, Historic building monitoring

Dedication (Optional)

First and foremost, we want to thank Almighty Allah for all the help, blessings, and guidance that have helped us along the way in our research. To our dear parents, whose unwavering support, sacrifices, and love made this possible. And to the great teachers at BRAC University, whose knowledge, help, and dedication have always been a source of inspiration and motivation.

Acknowledgement

We would like to thank our supervisor for always being there for us, giving us helpful advice, and supporting us throughout this research. We also want to thank our department for giving us the tools and resources we needed to do this work. Lastly, we want to thank our friends and family from the bottom of our hearts for always being there to support and encourage us on this journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Bangladesh boasts a rich cultural heritage that spans thousands of years, reflected in its diverse structures, religious sites, monuments, traditional architecture, and terracotta. These historic resources offer valuable insights into the country's rich artistic and social heritage, serving as priceless archives of cultural memory and identity. However, the vulnerability of these sites is increasing due to climatic factors (humidity, rainfall, temperature variation, air pollution), biological agents (microbial activity, fungal growth, vegetation), and last but not least, structural neglect [16]. Many of these cultural assets are at risk of gradual yet irreversible degradation due to the absence of systematic monitoring and intervention.

In Bangladesh, conventional preservation methods are mostly manual and time-consuming and rely on visual inspection by the respective authorities. Although expert surveys are essential, they are time-consuming, sensitive to subjectivity, and have limited scalability as well, especially while monitoring large numbers of heritage sites. Consequently, there is a desperate need for an automated, objective, and scalable solution for detecting early indicators of damage in cultural heritage sites.

Recent advancements in computer vision and artificial intelligence have revolutionized methodologies for visual identification tasks. Deep learning-based object detection models like YOLO (You Only Look Once) and Faster R-CNN have helped a lot in finding and categorizing objects in many fields, such as autonomous driving, medical imaging, and industrial imaging [4], [5], [17]. These frameworks have demonstrated promising potential in detecting subtle, localized anomalies within visual scenes. Using these models for heritage conservation can change how we keep an eye on structural integrity and surface damage, as well as give us ongoing, data-driven information about how to manage heritage.

1.2 Motivation

Machine learning (ML), deep learning (DL), and computer vision have been successfully applied to heritage sites like the Dunhuang Mogao Grottoes [13], Hampi, and Timurid architecture [14]. However, little is known about the usage of object

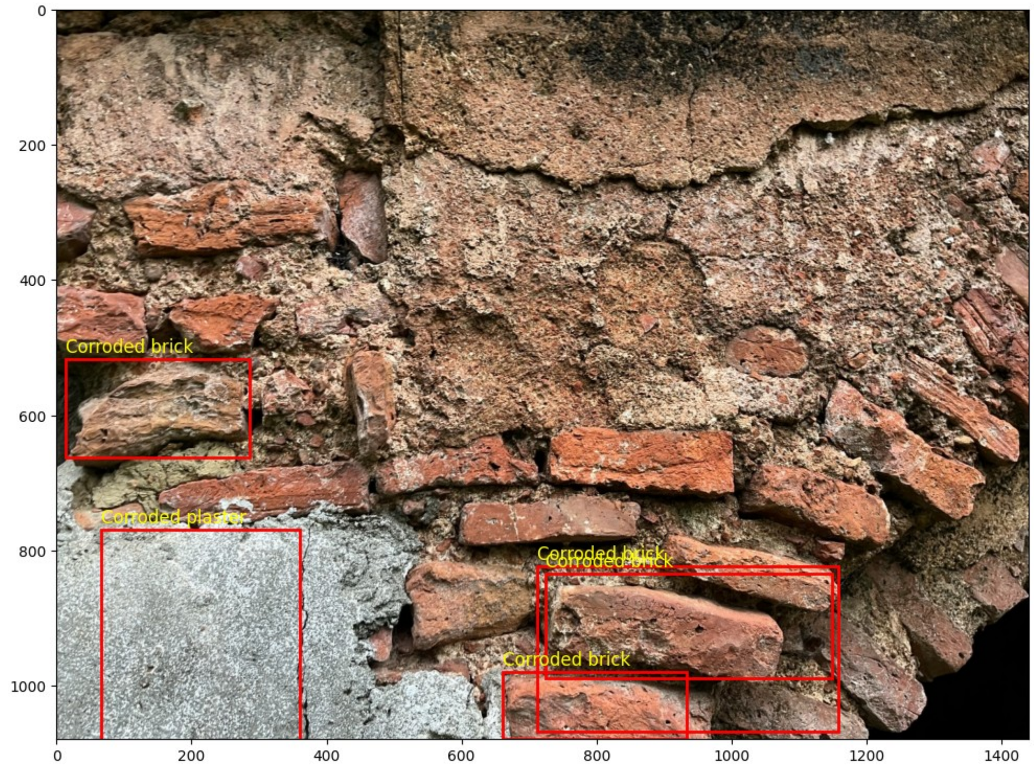


Figure 1.1: Sample image from the heritage dataset showing both Corroded Brick and Corroded Plaster damages [18]

detection models in cultural heritage preservation, especially for monitoring surface damage in historic structures.

This study proposes a framework for AI-powered damage identification in historic structures. By training object detection models, it aims to develop an automated approach to identify and classify surface deterioration, helping conservation specialists monitor and manage risks to heritage integrity. The considered damage categories are:

- Artistic
- Corroded plaster
- Corroded brick
- Fungus features
- Living plant features

A figure showing the combined damages of corroded brick and corroded plaster is presented in Figure 1.1.

Moreover, this study put emphasis on 2D image-based deep learning models to detect, classify and localize surface damages. Robustness against dataset imbalance is ensured through the adoption of data augmentation techniques, both offline augmentation using Albumentations library and online augmentation within YOLO training. Moreover, integrating ensemble learning through Weighted Boxes Fusion (WBF) enhances detection accuracy by combining the complementary strengths of YOLOv8 and Faster R-CNN.

Digital documentation and metadata organization require a lot of computing power, but AI makes it easier to keep things safe while also raising questions about scalability and ethics [21]. It is important to make sure that
AI is a helpful tool and not a replacement for human knowledge [22]. This project uses an ensemble-based deep learning method to find damage to heritage sites, making it a scalable, cheap, and fair alternative to traditional manual inspection. This study's results have direct consequences for cultural heritage management by facilitating early interventions, sustained monitoring, and digital archiving of structural conditions.

A historic image of Panam City is shown in Figure 1.2.

Figure 1.2: Panam City[25], a historical site with notable heritage structures.

1.3 Problem Statement

Deep learning is moving forward quickly, which opens up new ways to protect heritage through automated damage detection. This study investigates the identification and localization of diverse surface deterioration types in images of historic Bangladeshi buildings through contemporary object detection frameworks.

The study uses two cutting-edge models: YOLOv8, which is known for its accuracy and real-time performance, and Faster R-CNN, which is known for its speed. In addition, the use of Weighted Boxes Fusion (WBF) combines the predictions from both models to make an ensemble detection system that is as reliable as possible. YOLOv8 represents a current iteration of the YOLO paradigm, which is designed to ensure practical balance between speed, detection performance, and ease of use in real-life tasks, and Ultralytics maintains the official documentation and implementation details for this model [26]. Meanwhile, Faster R-CNN works by collecting the region proposals and then refining as well as classifying these proposals. Robust localization and classification accuracy are gained by this decomposition, particularly in situations involving small objects or subtle visual inputs. Hence, Faster R-CNN remains a fundamental reference for region proposal-based detection architecture [4]. Moreover, ensembling combines predictions from different models, which is an effective strategy to exploit complementarity. Weighted Box Fusion is a pragmatic post-processing approach that combines overlapping boundary boxes from multiple detectors through weighting them based on confidence, and it leads to more accurate and stable final boxes than NMS or naive non-max suppression in many cases [12].

"A Multi-format Open-Source Historic Wall Image Dataset", sourced from Mendeley Data(2023) was used for this research. The dataset contains 2292 RGB images (but only 340 images were originally labelled) which were captured from one of the most prominent heritage sites of Bangladesh, Panama City, Sonargaon. The dataset has five damage categories. However, due to class imbalance and data quality,we used 340 originally labeled images and generated 1,000 augmented images, for a total of 1,340 images used for training and evaluation.. Segmentation masks provided by the dataset were converted into bounding box annotation which is appropriate for object detection models.

The suggested framework aims to enhance accuracy and resilience by utilizing online augmentation during training to improve generalization and offline data augmentation to balance class frequencies. This approach enables conservationists to detect damages earlier, reduce reliance on subjective manual examination, build a framework for long-term heritage management, and represents an innovative digital solution for cultural heritage monitoring in Bangladesh.

1.4 Research Objectives

The main objective is to develop and evaluate an AI-driven framework for automated surface damage detection in Bangladeshi heritage structures using low-cost, scalable, and image-based techniques. The research aims to:

- **Dataset Preparation and Augmentation:** Organize and prepare the dataset, turn segmentation masks into bounding boxes, and use augmentation methods to fix the class imbalance.

- **Model Training:** Create and train object detection pipelines for YOLOv8 and Faster R-CNN.

- **Ensemble Integration:**For better reliability, use Weighted Boxes Fusion to combine the outputs of YOLOv8 and Faster R-CNN.

- **Evaluation:** Use mAP@0.5 and per-class AP to evaluate performance, and look at the trade-offs between recall, prediction, and inference speed.

1.5 Methodology in Brief

Four main steps in the structured, data-driven method is used:

- **Preparing and Adding to Data:** The dataset, which had five classes (Artistic, Corroded Plaster, Corroded Brick, Fungus, and Living Plants), was put together, changed to YOLO format, and added to in order to fix the data imbalance. The dataset was divided into three parts: 70% for training, 15% for testing, and 15% for validation.

- **Model Training:** The training of YOLOv8 and Faster R-CNN was done separately. We used the AdamW optimizer and different augmentations to fine-tune YOLOv8 for 100 epochs. Using SGD with momentum and a WeightedRandomSampler, Faster R-CNN was fine-tuned for 24 epochs.

- **Ensemble Integration:** WBF was used to combine the predictions from both models, using YOLO's recall and Faster R-CNN's accuracy.

- **Assessment:** Mean Average Precision (mAP) was used to measure performance. It was calculated as the area under the precision-recall curve.

The methodology using a flowchart is described in Figure 1.3.

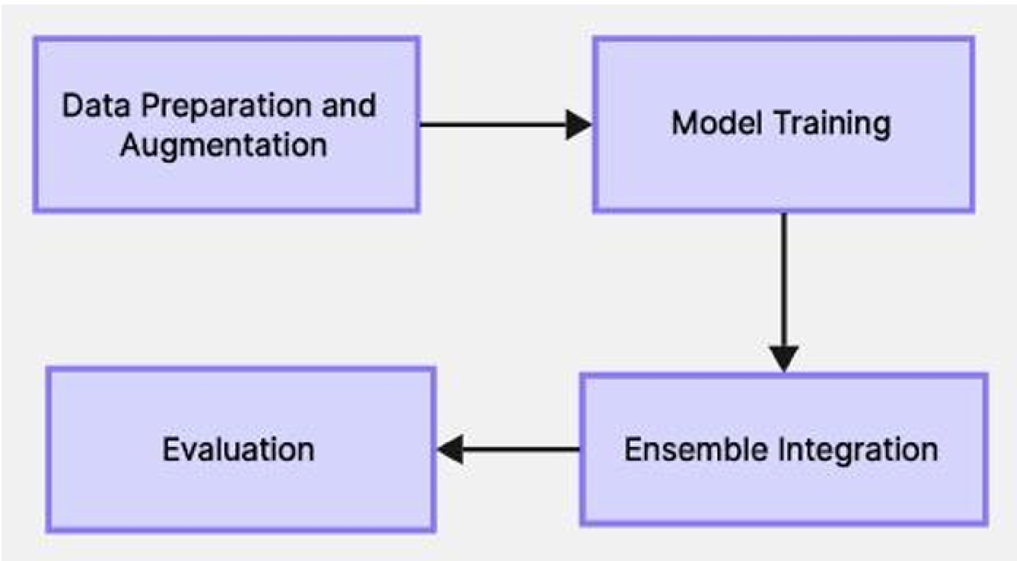


Figure 1.3: Overview of the proposed methodology for AI-based damage detection.

1.6 Scopes and Challenges

Bangladesh's cultural heritage sites are now at risk because of biological, environmental, and human-made pressures. The current inspection methods are expensive, take a long time, are based on personal opinion, and are not good enough for large-scale monitoring. AI-based image analysis is more promising than other methods. There are a number of areas where the research can make a big difference. But it also has a lot of trouble finding damage to heritage.

Scopes:

- **Cultural and Geographical Context:** Focus on the damage to the visual surfaces of Bangladesh's cultural heritage structures. The proposed architecture can be used to help protect heritage sites in other areas where there isn't much data.
- This framework can be adapted for the effective monitoring of museums, archaeological sites, and restoration organizations.

Research Contributions

- **Bridges two fields:** This work helps us connect digital preservation with computer vision (using AI) where heritage research and AI meet.
- **Sets a testing standard:** It provides a reliable way of testing deep learning models, especially when an optimum quantity of data is not available.
- **Boosts detection:** By adding an ensemble-based detection framework, it becomes more accurate and reliable at finding surface damage.
- **Shows key connections:** It demonstrates how factors like the model's design, the amount of available data, and the metrics used all work together to make the usage of "AI for preservation" a success.

Challenges

- **Complex Surfaces:** Unlike industrial settings, heritage structures are not always uniform. The same type of damage can look vastly different on various buildings depending on the lighting, angle, and surface material. This complexity requires models that can handle a wide variety of visual inputs.
- **Model Limitations:** Each model has a weakness. The Faster R-CNN is accurate but slow. YOLOv8 is fast but might have small localization errors. Using only one model means you risk either missing damage or getting the location slightly wrong.
- **Data Scarcity:** A major problem is the lack of labeled heritage datasets. While large general datasets like COCO exist, finding specific, labeled data on historic damage is difficult. It's hard to access heritage sites to take pictures of damaged surfaces which often leads to an imbalance where some types of damage are not fully represented.

Chapter 2

Literature Review

2.1 Preliminaries

This section lays the groundwork for understanding the heritage defect detection framework by explaining some basic ideas.

Image Representation and Bounding Boxes

Digital images are three-dimensional arrays (Height $\times$ Width $\times$ Channels) with pixel values normalized to $[0, 1]$. Bounding boxes represent object locations using formats: absolute coordinates (x_1, y_1, x_2, y_2) or YOLO format (c, x_c, y_c, w, h) with normalized values. Intersection over Union (IoU) quantifies box overlap:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2.1}$$

YOLOv8 Architecture and Algorithm

YOLOv8 **yolov8** employs a three-component architecture: the CSPDarknet backbone with partial Cross-Stage connections for hierarchical feature extraction at multiple scales (stride 8, 16, 32), Path Aggregate Network (PAN) neck combining top-down and bottom-up pathways for multi-scale feature fusion, and an anchor-free decoupled head separating classification and localization tasks. YOLOv8 uses task-aligned optimization algorithms to maximize three loss components during training: Binary Cross-Entropy for classification, Complete IoU for box regression, and Distribution Focal Loss for modeling uncertainty. The network processes images in a single forward pass during inference, predicting objectness scores, class probabilities, and bounding box coordinates for each grid cell. Then, Non-Maximum Suppression (NMS) is used to get rid of detections that are too close together.

Faster R-CNN Architecture and Algorithm

Faster R-CNN **fasterrcnn** implements a two-stage detection pipeline consisting of a ResNet-50 FPN backbone with residual connections and feature pyramids for multi-scale representation, Region Proposal Network (RPN) using anchor boxes at multiple scales (128^2, 256^2, 512^2) and aspect ratios (1:1, 1:2, 2:1) to predict objectness and refinement offsets, ROI Align pooling extracting fixed-size features from variable proposals via bilinear interpolation, and detection head with fully connected layers predicting refined class probabilities and class-specific box adjustments. Training employs joint optimization with positive samples (IoU ≥ 0.7) and negative samples (IoU < 0.3) for balanced learning. During inference, the backbone generates feature maps, the RPN proposes regions, the detection head refines predictions, and NMS eliminates redundant detections to produce final outputs.

Loss Functions and Optimization

Object detection optimizes combined objectives:

$$\mathcal{L} * \text{total} = \alpha\mathcal{L} * \text{cls} + \beta\mathcal{L}_{\text{loc}} \tag{2.2}$$

Training uses SGD with momentum or AdamW optimizers. Learning rate schedules (cosine annealing, step decay) enable rapid initial learning followed by fine-tuning.

Transfer Learning

Pretrained models (COCO: 330K images, 80 classes) provide learned feature extractors. Fine-tuning adapts these weights to heritage defect detection with limited data, reducing computational cost and improving generalization.

Data Augmentation

Geometric transformations (flip, rotation ±10%, translation 15%, scaling 70-130%) and photometric adjustments (brightness, contrast, CLAHE, noise, blur) increase dataset diversity. Advanced techniques include Mosaic (4-image grids), Mixup (image blending), and Copy-Paste (object transfer). Bounding boxes are updated accordingly with a 25% minimum visibility threshold. Augmentation attempts were configured to over-sample minority classes.

Evaluation Metrics

Precision and recall measure detection quality:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN} \tag{2.3}$$

Average Precision (AP) computes the area under the Precision-Recall curve via 11-point interpolation. Mean Average Precision (mAP@0.5) averages AP across classes at IoU $= 0.5$. mAP@0.5:0.95 averages across thresholds $[0.5, 0.95]$ for stricter evaluation.

Class Imbalance Mitigation

Oversampling generates additional minority class instances. Weighted sampling adjusts probability by inverse frequency:

$$w_i = \frac{1}{f_i} \text{ where } f_i = \frac{n_i}{\sum_j n_j} \tag{2.4}$$

Weighted Boxes Fusion

WBF **wbf** combines predictions from multiple models by clustering overlapping boxes (IoU > 0.55), computing weighted coordinate averages, and fusing confidence scores. Model weights (e.g., 2:1 YOLO:Faster R-CNN) reflect relative performance, typically improving accuracy by 1-3%.

Dataset Organization

A standard 70/15/15 train/validation/test split with random shuffling ensures balanced class distribution. Validation monitors overfitting, while testing provides unbiased evaluation. Strict separation prevents data leakage. The dataset was imbalanced toward classes like Fungus and Living plant, which influenced model performance.

Computing Infrastructure

CUDA-enabled GPUs provide 10-100× speedup through parallel matrix operations. Image caching loads datasets into RAM once, achieving 2-3× training acceleration. Batch processing maximizes GPU utilization.

2.2 Review of Existing Research

The authors in [1] wanted to fix a broken terracotta statue called the Madonna of Pietranico that was damaged by an earthquake in 2009. They used 3D scanning, putting things back together in real life, making things look like they were painted in real life, and putting things back together in virtual reality to guess how the pieces fit together and make structural supports. Meshlab was used for virtual painting and digital reassembly, and rapid prototyping made it possible to make exact physical supports. The restoration also looked into tough issues, like how to recreate lost polychrome decorations using old photos, extra evidence, and theories about how to put things together. This adaptable method shows how digital tools can help protect cultural heritage.

The authors of [2] provide a comprehensive examination of 3D building modeling techniques utilizing LiDAR data and imagery. It breaks the process down into three parts: using images to rebuild something, using LiDAR to model something, and using both methods together through sensor fusion. It talks about important issues like cutting down on noise, recording data, and upsampling to get more accurate results. Some of the models that have been used are Multi-View Stereo (MVS), Structure from Motion (SfM), and segmentation based on LiDAR. It has also talked about how automated and interactive modeling techniques are getting better. This shows how much people want high-resolution 3D city models for things like navigation and virtual reality.

The author said [3], To restore historical works in a virtual way, digital image processing is very important. It tries to help us find cracks and fill them in, which is important for both keeping the artwork's visual integrity and for art historical analysis. Cornelis et al. (2012) put forth a complete system for finding and fixing flaws in digitized paintings, with the 15th-century Ghent Altarpiece as their main focus. Their answer has three different detection algorithms that work well together to find cracks of all sizes, shapes, and brightness. Then, to cut down on false positives, they use a semi-supervised clustering stage. This ensemble method makes it easier to find cracks in painting areas that are hard to see, like brushstrokes. During the inpainting stage, a patch-based method is used to fill in the parts of the image that are missing without changing its structure or texture. The authors illustrate the efficacy of this methodology via a case study that improves the legibility of historical text within the artwork, exemplifying how digital restoration can facilitate paleographic research and conservation initiatives. This study underscores the efficacy of integrating advanced detection algorithms with specialized inpainting techniques to enhance restoration quality, representing a significant advancement in the domain of cultural property preservation.

The writer of [4] says that combining deep learning with region proposal methods has had a big impact on object identification. Faster R-CNN is a big step forward in this area. Selective Search and EdgeBoxes were two old methods that used low-level properties to suggest regions, but they were slow and hard to use on a large scale. Faster R-CNN's Region Proposal Networks (RPNs) fixed this by using the same convolutional features in both the proposal and detection stages. This made it possible to train from start to finish and almost in real time. The RPN makes good suggestions while keeping translation invariance and simplifying the model by using anchor boxes of different sizes and aspect ratios. Faster R-CNN was much faster and more accurate than older methods when tested on benchmarks like PASCAL VOC and MS COCO. It worked best with fewer recommendations for regions. This new idea led to modern object detection pipelines and has had an effect on more research in areas like 3D detection, image captioning, and segmentation.

Redmon et al. (2016) in [5] presented YOLO (You Only Look Once), an innovative approach for object detection that conceptualizes the task as a singular regression problem, predicting bounding boxes and class probabilities directly from complete images in a single evaluation. YOLO is different from older methods that used image classifiers to find things because it has a single, end-to-end trainable architecture that makes both localization and classification better at the same time. This speeds up and improves the process. It can find things in real time by processing images at 45 frames per second. Fast YOLO, a smaller version, can do this at speeds of up to 155 frames per second. This is a lot faster and more accurate than real-time detectors that came before it. YOLO makes more mistakes when it comes to localization than region-based detectors like R-CNN. However, it has a lower false positive rate on background regions and works well in many different areas. It can work well with non-natural image datasets, like artwork, which shows how adaptable it is. This work has laid the groundwork for better object detection architectures in the future. A lot of real-time detection models have been changed by it, and it is now an important part of many practical applications, like self-driving cars, surveillance, and robotics.

In [6], Historically, methods like terrestrial laser scanning, photogrammetry, and radar interferometry have been used to inspect historic buildings. However, these methods are expensive, time-consuming, and sometimes not possible in places that are hard to get to. Visual inspection is still common but subjective and prone to human error, prompting the development of automated systems using robots and UAVs for image collection. Earlier crack detection techniques based on handcrafted features (e.g., edge detection, percolation, and fractal methods) often struggled with noise and complex patterns, while machine learning approaches such as SVMs offered improvements but were limited by feature extraction accuracy. Recent advances in deep learning, especially Convolutional Neural Networks (CNNs), enable automatic feature learning from raw images, outperforming traditional methods in robustness and accuracy, and when combined with classifiers like SVMs, they show even higher performance, making CNN-based systems highly effective for automated crack detection and monitoring of heritage structures.

In [7],The author presented the integration of 3D modelling, photogrammetry and augmented reality (AR) to reconstruct and visualise fragmented sculptures. For this purpose they have worked on a Roman lion statue which was damaged. By using photogrammetric technique the reconstruction of the damaged sculpture was done. To provide some missing parts, photogrammetry on the other sculptor was also done. In order to do photogrammetry, a software named Recap was used which can process at max 100 photos. After integration of all these parts using 3D modelling the reconstruction of missing detail was performed. 3D modelling was done by a professional software PhotoScan (Agisoft) which used 130 photos and generated a more detailed texture model of a lion. Finally, using the Unity engine platform, the real model was augmented.

In [8], the authors present a comprehensive study on the application of segmentation and classification techniques for 3D digital heritage preservation. The authors focus on automating the classification of complex 3D data, such as point clouds and polygonal mesh models, using supervised and unsupervised machine learning methods. They discuss the challenges of segmenting and classifying heritage assets, given their complex geometries and textures. The study uses these methods on different archaeological and architectural situations to show how well they work for restoring, analyzing, and documenting heritage sites.Tests done on old buildings like the Pecile's wall and Bologna's porticoes show that random forest algorithms, especially, are very good at classifying things, especially when it comes to finding architectural features and restoration zones. The research indicates that the suggested methodologies are dependable; however, enhancements are necessary for optimal functionality in more complex heritage contexts.

In [10],The authors give a full overview of how machine learning (ML) can be used to protect and study cultural heritage (CH). They talk about important machine learning methods, such as supervised, semi-supervised, and unsupervised learning. They also talk about how deep learning techniques have been changed to work with tasks related to cultural heritage, such as putting artifacts back together, sorting images, and grouping data. The authors talk about the problems that CH faces, especially because it is hard to get to big, high-quality datasets. The paper stresses how important transfer learning is, which is when models that have already been trained are changed to fit tasks that are specific to CH. Convolutional neural networks (CNNs) and support vector machines (SVMs) have shown potential in classifying art forms, identifying artists, and detecting materials used in ancient artifacts. Fiorucci et al. say that even though these things have happened, ML use in CH is still in its early stages. They say that more research needs to be done to make datasets more available and to improve ML methods even more in order to suit the complex needs of cultural preservation.

For [11], The authors present a distinctive viewpoint on the reconstruction of 3D objects from a single 2D image in their work. To address these problems, the authors propose a model that utilizes a Convolutional Neural Network (CNN) to convert point clouds into 3D object representations. They also came up with a way to take a point cloud that is shaped like a sphere. They said that this would make all the points (the surface of the object) completely uniform, which would guarantee the best quality of the final reconstruction to make the true and detailed 3D model. Finally, they also used metrics like the Chamfer Distance and Earth Mover's Distance to validate their model on both synthetic and real datasets, ShapeNet and Pix3D, respectively. The model did do better than some older ones, including Point Set Generation Network (PSGN) and 3D-LMNet, but it did have to work a little harder to put thin or narrow pieces of the object back together. This study shows that the proposed network model with an initial point cloud is a reasonable way to do 3D reconstruction. However, it needs to be improved for geometrical objects with complex geometric features.

In [12], Recent progress in computer vision has made it possible to utilize deep learning to automatically find deterioration in cultural heritage buildings. This solves problems with manual inspection, such as bias, high labor costs, and safety issues. Pathak et al. (2021) developed a novel methodology that combines 2D visual data and 3D point clouds to detect surface issues, such as cracks and spalling, on cultural sites using deep learning models. They use semantic segmentation to prepare the data, Faster R-CNN to find objects, and 3D models that match the objects to show damage. This gives you a sense of space and makes it easier to make plans for preservation. The curated Ayutthaya dataset, which had pictures of old buildings that were broken, helped the model learn how to find things by looking for features that don't change with the background. It worked well even with new data, like photos of Hampi, a UNESCO World Heritage Site. The research demonstrates that deep learning can monitor cultural assets without causing harm and is scalable. This is a good start for future work on keeping digital files safe and checking the health of buildings.

The authors present a project utilizing artificial intelligence to safeguard the Dunhuang Mogao Grottoes, an essential component of World Heritage, as detailed in [13]. The paper examines a multifaceted methodology that employs AI in the preservation, restoration, and analysis of ancient grottoes that have withstood a millennium of detrimental degradation. To automate massive scale mural restorations, the authors propose deep networks and their outcome is arguably much better than what can be achieved via manual practices by archaeologists, while the process itself becomes much faster. AI has also been used in object detection and retrieval (when it's logistically impossible to label them by hand) and will offer tremendous help in those cases. Finally, many of these advancements represent good breakthroughs, specifically in the use of style transfer technology to trace artistic styles between various time periods, enabling new ways of looking at and interpreting art across eras. Restoring 10,000 images, 3455 for style transfer, 6147 for property retrieval, are contained in the AI dataset for future research based grounds. The authors performed a very careful evaluation and comparison of cutting edge techniques, and presented a thorough analysis on how effectively they performed.

In [14], the authors take a data-driven approach, using Machine Learning (ML), Deep Learning (DL) and computer vision etc., to provide an overview on digital restoration techniques about our cultural heritage. They outline how emerging technologies such as high-precision photogrammetry and laser scanning are improving traditional restoration processes of heritage sites today. Techniques like image inpainting (filling in the damaged areas of works of art) are now being used in projects related to the restoration of Hampi and Timurid architecture. It also explores learning-based techniques such as Generative Adversarial Networks (GANs), with examples of how these methods have surpassed earlier ones in the task of 3D reconstruction. Crowdsourcing as well as time-lapse shooting tackle specific challenges about how to obtain and process data. It concludes with promising advances for IoT sensor data collection, new interaction methods extended via augmented reality, and performance improvements for using artificial intelligence in virtual restoration environments. Nonetheless, this article recognises the need for greater research to bridge model accuracy and scalability gaps for wider use.

In [15], The authors examine the use of AI and Machine Learning (ML) in the conservation and restoration of cultural assets, primarily focusing on virtual restoration. They say that natural and man-made things (including war, pollution, and changes to the environment) will hurt tangible cultural assets, such works of art and old structures. But traditional manual restoration procedures work, but they take a lot of time, are subjective, and cost a lot. We think that deep learning models are a better option than traditional manual approaches since they are easier to use, more scalable, and more accurate. AI can automate methods for repairing or filling in missing or damaged elements of a work of art (inpainting).Second, they show models that look promising for restoring images or 3D structures by looking at high-resolution scans and making believable reconstructions. These models are called Mask R-CNN (Region-based Convolutional Neural Network) and Wasserstein GAN (Generative Adversarial Network). They also warn about the ethical effects of AI on cultural preservation to avoid bias, since it changes the original objects. Finally, the authors say that AI makes it easier for people to save their cultural heritage by giving them new ways to restore, reach, and keep history for future generations.

In [19], The authors examine the role of AI in safeguarding cultural assets through technological means. The authors assert that AI possesses the capacity to transform the subsequent conservation processes: digitalization, documentation, restoration, and environmental monitoring. Akyol and Avcı talk about how new technologies like automatic scanning, 3D modeling, and virtual reality are making it possible to make exact digital copies of historical sites and objects that are available to scholars, teachers, and the general public. It also talks about how important AI algorithms are for figuring out how bad the damage is and how to fix it by finding patterns and recognizing images. You can use AI to keep an eye on the environment and find and fix problems like changes in temperature and humidity. You can also use it to protect heritage sites before they are damaged. The article talks about how AI can help keep cultural heritage safe. It also talks about the ethical problems that come up with these technologies and makes the case for finding a balance between AI and the historical integrity of heritage assets. It also points out the clear benefits. Scientists say that AI can be a big help in protecting cultural heritage and making sure it lasts for a long time if it is used correctly.

In [21],The authors investigate the increasing utilization of AI technology in the conservation and restoration of cultural heritage. In this study, the authors analyze the influence of AI-driven innovations, such as generative AI, digital twin mapping, and machine learning, on traditional conservation methodologies. The literature examined in the study outlines the advantages and disadvantages of AI, emphasizing improved accuracy in digital archiving and virtual reconstructions, while also addressing technical challenges and ethical issues, especially in relation to cultural authenticity. The study involves a varied group of professionals—such as conservationists, archaeologists, architects, and technologists—through interviews to clarify the complex discussion surrounding the use of AI in preventing structural damage and enabling predictive maintenance, while also highlighting the shortcomings of AI in measuring the historical and artisanal elements crucial for genuine restoration. There are also moral questions that come up naturally, like how much you can give up tradition to keep a machine running well. The paper concludes that, if properly integrated, AI has the potential to significantly transform the field, provided that the technology is aligned with and advances the core objectives of heritage preservation.

In [23], Object detection is a very important task in computer vision, which has progressed better than traditional features, like detectors like Viola-Jones and many downgraded models to deep learning-based approaches such as R-CNN, Fast R-CNN, and Faster R-CNN, significantly improving accuracy at considerable computational expense. One-stage detectors like YOLO and SSD were made to make real-time performance possible. These detectors find a good balance between speed and accuracy. Detection got even better thanks to new technologies like RetinaNet's focal loss and DETR's transformer-based architecture. The YOLO family has made a lot of progress in a short amount of time. For instance, YOLOv8 added anchor-free detection, decoupled heads, and flexible deployment options, which made it good for tasks like pose estimation and segmentation. Adding spatial information to RGB data by combining depth sensors like the Intel RealSense D435 with object detection systems has also made them more accurate in difficult situations. This has improved performance, distance estimation, and localization in areas like robotics, self-driving cars, and augmented reality.

The author described that [24],Recent advances in deep learning have made it much easier to protect cultural heritage by making it possible to automatically find surface damage. Traditional methods of inspection by hand and image-based techniques that use handcrafted features often have problems with speed, accuracy, and consistency. Older methods that used classical machine learning, such as SVMs, worked better, but they still needed a lot of work to add features. Real-time and reliable damage detection became possible for complex heritage environments with the creation of object detection frameworks like YOLO. With each new version, YOLO has gotten better. The newest version, YOLOv10, is light, fast, and accurate. This is great for finding cracks, surface loss, and other types of damage in old buildings. YOLOv10 is better than CNN-based classifiers and segmentation models like Mask R-CNN at finding the right balance between speed and accuracy. This makes it a good tool for long-term and scalable monitoring of the health of structures in heritage conservation.

2.3 Summary of Key Findings

The literature on digital technologies based on the restoration of the cultural heritage works One several trends . First of all, a massive shift from manual inspection and features which Are handcrafted in the purpose of doing pipelining using the algorithms like Faster R-CNN, YOLO and CNN. They keep outperforming edge based and classical machine learning baselines for example SVM, K-NN through huge noise with having robustness and cross Site generalization [5], [20], [21], [22].

Growing multi dimensional integration that combines 3D technology like photogrammetry, LIDAR, point clouds using 2D image analysis, visualization of Augmented reality and virtual reality and IOT sensing for monitoring the real time structures for the assessments [15], [16]; The data which are persistent results in data scarcity relative to many general purpose corporations which are COCO, ImageNET . They are addressed through some augmentation and transfer learning in spite of having a larger domain sets remaining exceptions [5], [9]; and increasing use of hybrid/ensemble detectors—pairing YOLO's speed/recall with Faster R-CNN's precision and reconciling outputs via Weighted Boxes Fusion—to improve reliability across defect types and imaging conditions [11].
Along with these improvements, there is a strong emphasis on ethical and methodological guardrails: AI should not take the place of expert stewardship; it should only improve it. It should also reduce bias, protect authenticity, and require clear documentation with human oversight [14], [19], [21]. In this context, an ensemble of YOLOv8 and Faster R-CNN with WBF is well-justified for heritage scenarios with limited data. However, the field necessitates more stringent evaluation protocols—definitive annotation standards, transparent dataset-limitation disclosures, and cross-site validation across South Asian architectural typologies—to guarantee that AI-enabled preservation is both effective and respectful of cultural integrity.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

Problem Framing and Scope
The goal of this study is to create a system that can find and categorize different kinds of flaws on heritage surfaces using RGB images. The system can tell the difference between Artistic, Corroded Brick, Corroded Plaster, Fungus, and Living Plant by looking at pictures of old buildings.

The implementation brings together two models for detecting objects:

- YOLOv8, a one-stage detector that works in real time.
- Faster R-CNN with ResNet-50-FPN, which is known for being very accurate.

YOLOv8 is a faster way to find things, and Faster R-CNN is a more accurate way. Using Weighted Boxes Fusion (WBF), which combines overlapping detections based on confidence weighting, makes the models' predictions even more accurate. The architecture can work with limited data and hardware thanks to the combination of YOLOv8 and Faster R-CNN, which keeps both high accuracy and real-time detection.

Functional Requirements

- **Data Collection and Curation:**
  - It takes RGB images in PNG and JPEG formats with a minimum resolution of 1024 pixels on the shorter side. Right now, it is resized to 768 for YOLO.
  - Supports annotation in YOLO (.txt) formats and COCO formats along with incorporated tools for transformation
  - Integrates project-level metadata, including capture date, camera type, and location, to help in heritage data organization.

- **Annotation Protocol:**
  - Every image is annotated with bounding boxes that mark the clear instances of five target categories.
  - Double annotation of less than 10% with arbitration is a quality policy (planned as the dataset scales).
  - Class consistency and inter-annotator agreement are audited when human labeling occurs.
- **Model Training and Evaluation:**
  - Models—YOLOv8 and Faster R-CNN (ResNet-50-FPN).
  - Augmentation technique for
    (i) offline: Horizontal flip, RandomBrightnessContrast, HueSaturationValue, CLAHE, GaussNoise, MotionBlur, Affine (scale/translate/rotate/shear);
    (ii) online (YOLO): mosaic, mixup, copy-paste, RandAugment.
  - Class imbalance is handled via targeted offline oversampling of minority classes and a WeightedRandomSampler for Faster R-CNN.
  - During inference, Weighted Boxes Fusion (WBF) is utilized for ensemble integration.
  - Evaluation metrics include mAP@0.5, mAP@0.5:0.9, precision-recall, confusion matrices and curves.
  - Site- and facade-based splits manage validation to prevent data leakage.
- **Inference and Reporting:**
  - Batch inference is supported on both CPU and GPU.
  - Outputs contain bounding box coordinates that are normalized and exported in CSV/JSON format.
  - Measured site-level summaries and visualization overlays are automatically generated for reporting purposes.
- **MLOps & Reproducibility:**
  - Deterministic seeds, pinned configurations, and run logs are maintained. Versioning of datasets, code, parameters, and weights is targeted for releases; model cards and dataset datasheets are planned.

Non-Functional Requirements

- **Target Performance:** Using a re-annotated, class-balanced dataset, we target an mAP of at least 0.60 overall and 0.50 for each class, with a temporary goal of mAP@[0.50:0.95] of at least 0.35.
- **Runtime:** For 1280×1280 inputs (YOLO variant), inference takes less than 150 ms/image on a T4-class GPU and less than 1 s/image on a CPU-only system at 640×640.

- **Maintainability:**Explicit configuration files (YAML), single-command training and evaluation, deterministic seeds and visualizations.
- **Security & Governance:** Allows image licensing and hides sensitive geolocation information when needed.

Requirement Achieved from Current Results
The first tests showed that most of the classes don't have bounding boxes, except for the Artistic class. Automatically generated bounding boxes worked well for validation but not so well for generalization in practice. So, the following things must happen in the future:

- Re-annotate at least 1000 more images with bounding boxes.
- Classified dataset splits by site, date of capture, and facade.
- Use audited augmentation, focal loss, and hard negative mining to make detection more accurate.
- Full documentation of how to set up the model and pre-process the data so that it can be reproduced.

3.2 Societal Impact

Positive Effects
The study presents cost-effective and scalable techniques for heritage preservation. It can find things early, which also helps with the following:

- Lowers long-term maintenance costs and makes repairs possible on time.
- Supports grant applications to make sure that custodians with less money have the same access to diagnostics.
- Helps protect endangered heritage buildings and digital files for the long term.

Potential Risks and Mitigations

- **Automation Bias:** It is mitigated by combining human-in-the-loop review and visualization uncertainty in inference output.
- **Vulnerable Site Exposure:** It is addressed by phased disclosure along with geolocation masking in public datasets.
- **Coverage Inequality:** Inequity of coverage is reduced by compiling diverse datasets and providing performance reports by region.

3.3 Environmental Impact

Energy and Emissions of Carbon
Deep-learning models absorb significant energy while training. These emissions depend on hardware type, training duration, dataset size, and power grid efficiency. This research uses YOLOv8 and Faster R-CNN, both of which are suitable for experimentation under limited resources. To enumerate environmental cost, the MLCO2 impact tool is used to evaluate and report the carbon footprint for every experiment. [1]

Mitigations

- Application of compact architectures such as YOLOv8 and Faster R-CNN instead of large-scale variants.
- To reduce unnecessary computation, applying early stopping and curve extrapolation.
- Maintenance of energy budget as well as tracking emissions per experiment.
- Execution of scheduling in low-carbon regions while using cloud resources.

3.4 Ethical Issue

- **Data Rights & Licensing:** The dataset used is a multi-format Open Source Historic Wall Image Dataset, sourced from Mendeley data. It is publicly available and under an open academic license. Every annotation and label derived in the research complies with the dataset's original usage terms.
- **Privacy & Provenance:** Exact coordinates and metadata are not disclosed without permission to protect heritage sites.
- **Bias & Fairness Metrics:** Metrics are reported by class and site. Dataset diversity is expanded by capturing different materials, lighting conditions, and damage types. Lastly, evaluation results are audited for bias to ensure that minority classes are not underrepresented.
- **Transparency & Accountability:** Each model release includes a Model Card and Datasheet explaining use, limitations, performance, and environmental impact.

3.5 Standard

This research upholds a consistent set of manually enforced standards across data, metadata, and evaluation protocols to ensure coherence, reliability, and interoperability. These standards serve as constraints, anchoring to maintain scientific rigor, facilitate future expansion, and promote widespread adoption.

- **Vision Data and Evaluation Standards:**
  - **Annotation Format:** The study applies YOLO/COCO-style bounding box annotations as the canonical format. Compatibility with most modern object detection architectures motivated this choice. This permits direct utilization of existing frameworks and evaluation code without custom translation layers as well.
  - **Evaluation Metric:** Model performance evaluation is done using mean Average Precision at IOU threshold 0.5, following the COCO-style evaluation paradigm. This metric is widely used in object detection-based research and allows direct comparison with published models. This standard ensures balanced precision and recall while capturing false-positive-negative trade-offs in a single scalar metric.
  - **Macro-Averaging across Classes:** The architecture employs macro-averaged mAP over the intended damage categories to reflect the multi-class ambition of the primary problem. Macro-averaging ensures each class is weighted equally in the final score regardless of the absence of some classes in the dataset slice.
- **Cultural-Heritage Metadata Standards:**
  - **CIDOC-CRM (ISO 21127:2014) standard:** CIDOC Conceptual Reference Model is an established ontology for cultural heritage information, which is also recognized as an international standard. The goal is to provide semantic interoperability and enable the mapping of heterogeneous cultural heritage datasets into logical conceptual schema. The metadata scheme used by the study optionally aligns with CIDOC-CRM.
  - **IIF or International Image Interoperability Framework:** IIF is widely used in the modern heritage and library sector. Incorporating IIF in the architecture ensures that image data and visual output are accessible, scalable, and integrable with external systems. This research proposes the delivery of image assets through IIF-compliant services to ensure standardization of viewing, annotation, and tiling of images.
- **Open Data Stewardship and Fairness Principles:** This research adopts the 'Fair Data Principles' guidelines to ensure data is findable, accessible, interoperable, and reusable.
  - **Findable:** For the sake of searching and indexing, all datasets, annotations, and derived assets will be assigned permanent identifiers such as UUIDs or DOIs and extensive metadata descriptions.
  - **Accessible:** Data and metadata will be accessible through conventional protocols like HTTP with authentication or open APIs.
  - **Interoperable:** To support semantic alignment and system integration, metadata vocabularies will stick to community standards like CIDOC-CRM, JSON-LD, IIF.
  - **Reusable:** Datasets will include versioning history, explicit utilization of license, and provenance details facilitating confident downstream reuse.

To enhance reproducibility, decrease duplication effort, and encourage the integration of datasets into global heritage research ecosystems, the adoption of FAIR practices is necessary. It has been proven that FAIR-based stewardship cuts costs and enables efficient reuse.

3.6 Project Management plan and Risk Management Plan

Our project management plan is shown in Table 3.1, and our risk management plan is shown in Table 3.2.

Work Package	**Work Description**	**Duration**
Data Management and Re-annotation	Dataset Collection, bounding box creation and metadata standardization	Month 1-3
Baseline Training	Train both models	Month 2-4
Ablations	YOLOv8 and Faster R-CNN separately and tune hyperparameters	Month 2-4
Ensembling and Calibration	Apply Weighted Boxes Fusion as well as ensemble weighting	Month 2-4
Field Validation	Deploy model on different heritage sites to evaluate the model	Month 5-6
Release and Integration	Prepare datasets, model cards, and documentation for open access	Month 6

Table 3.1: Project Management Plan

Risk	**Label**	**Impact**	**Mitigation Strategy**
Insufficiency/Imbalance	High	High	Re-annotation, classified partitioning, focal/class balanced loss, targeted sampling
Overfitting via Leakage	Medium	High	Segregate by location/facade/database segregation; identify duplicates; retain external validation set
Augmentation	Medium	Medium	Domain-validate ML, Domain harms realism of augmentation ranges; visual assessments
Compute Limits	Medium	Medium	Reduced backbones; premature cessation; meticulous ablations
Legal/Licensing Issues	Low-Medium	High	License examination; limited dissemination; custodial contracts
Sensitive-site Exposure	Low-Medium	High	Graduated disclosure; geospatial blurring for public releases
Stakeholder Non-Adoption	Medium	Medium	Co-design evaluations; elucidated results; instructional resources

Table 3.2: Risk Management and Mitigation Strategies

3.7 Economic Analysis

Cost Components

All direct costs related to data preparation, training, compute resources, and post-processing fall under the economic evaluation. The estimation below relies on market-average prices for Google Colab Pro, AWS EC2 (T4 GPU), and academic labor equivalency (in the context of Bangladeshi universities). The estimated cost for various components is shown in the Table 3.3.

Component	**Detail**	**Estimated Cost (BDT)**
Data Management and Annotation	Manual review, labeling, and metadata tagging (1600 images × $0.25/image)	44,000
Offline Augmentation	Storage overhead + local compute time	3,300
YOLOv8 Training (Cloud GPU)	8 hrs × $0.35/hr (T4 GPU)	308
Faster R-CNN Training	10 hrs × $0.35/hr	385
Ensemble Evaluation (CPU + GPU)	3 hrs × $0.30/hr	297
Result Visualization and Logging	Python processing, reporting, exporting	1,100
Documentation, Versioning, and Release Prep	Cleaning dataset, write-up, and model cards	5,500
Cloud Storage and Backup	10GB × $0.02/GB/month × 6 months	132
Total Estimated Cost		**54,824**

Table 3.3: Estimated costs for various components of the project

Comparative Cost Perspective

The Comparative cost analysis is shown in the Table 3.4.

Method	**Average Cost (BDT)**	**Efficiency and Scalability**
Manual inspection by experts (per location visit)	11,000 – 22,000	Impressive accuracy yet slow, low scalability, Low
Proposed deep-learning-based framework (per-run)	¡ 550	High consistency, high scalability, High
Full Project (Development cost)	55,000	Reusable and extensible

Table 3.4: Comparative cost analysis between manual inspection and proposed deep-learning framework

Return on Investment (ROI) Analysis

We evaluate ROI based on model accuracy, time saved per inspection, number of new faults detected, and expensive failures avoided. The following parameters were used:

$$Ct = 498.40 \times 110 \approx 54,824\,\text{BDT (Total development cost)}$$

$$Sp = 80\,\text{hours saved per year at 10 USD/hour} \times 110 \approx 88,000\,\text{BDT}$$

$$Rf = 1000 \times 110 \approx 110,000\,\text{BDT (Return from the project)}$$

The formula for ROI is:

$$\text{ROI} = \left(\frac{(Sp + Rf) - Ct}{Ct}\right) \times 100$$

$$\text{ROI} \approx \frac{(88,000 + 110,000) - 54,824}{54,824} \times 100 \approx 360.4\%$$

Therefore, the architecture demonstrates substantial cost-efficiency and long-term sustainability by yielding an estimated ROI of approximately **360%** within the first operational year.

Chapter 4

Proposed Methodology

This chapter gives a full and in-depth look at the systematic method used to create, put into action, and test a deep learning pipeline for finding and classifying different types of damage and features on old buildings. The research is based on a multi-stage, comparative framework. It starts with careful data preparation and augmentation and then moves on to the parallel training of two different state-of-the-art object detection architectures. The process ends with the combination of these models into a better ensemble, which is then put through a lot of testing to see how well it works.

4.1 Design Process and Methodology Overview

The primary aim of this research is to create a highly precise and resilient computational model for the automated identification and categorization of five specific categories of interest on historic buildings: "Artistic" features, "Corroded brick," "Corroded plaster," "Fungus," and "Living plant." The overall design is meant to do two things: create a model that works well and compare different deep learning methods for this specific problem area.
Figure 4.1 shows the workflow diagram for using YOLOv8 and Faster R-CNN to find damage. The research methodology is systematically divided into the following principal phases:

1. **Data Curation and Strategic Augmentation:** We first had to thoroughly analyze the original dataset to see how the different object classes were spread out. Since real-world data often isn't balanced, we focused on using a specific offline data augmentation strategy. The idea was to artificially increase the number of examples for the less common, "minority" classes. This helps create a more balanced and varied dataset, which prevents the model from becoming biased and also makes sure it works well on new, unseen data.

2. **Structuring a Dataset in a Formal Way:** After the augmentation process, the full, improved set of images is officially divided into three groups: training (70%), validation (15%), and testing (15%). This strict, tiered separation is very important for the research's integrity. It makes sure that the final model

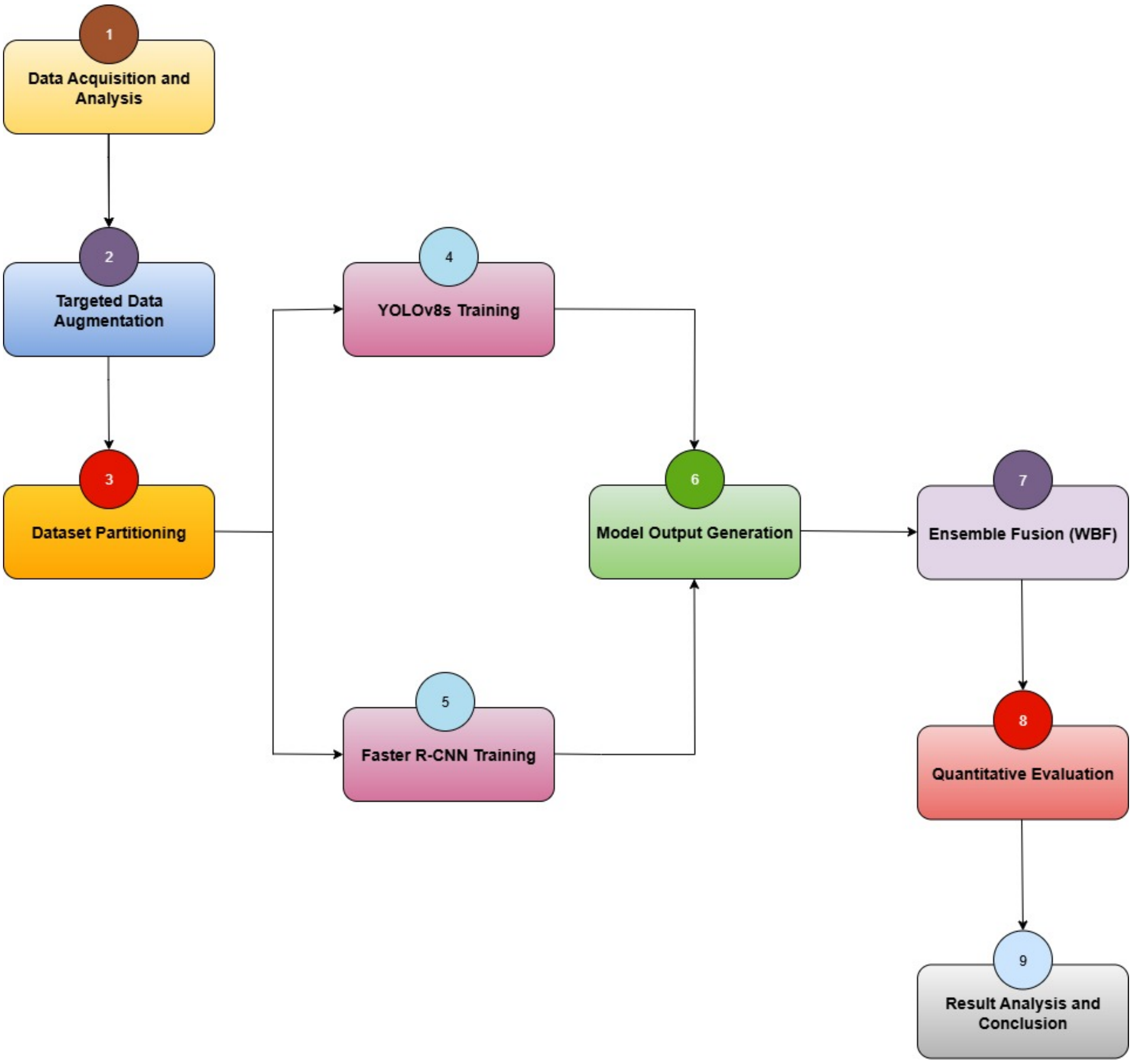


Figure 4.1: Workflow of the damage detection using YOLOv8 and Faster R-CNN

is tested on data it has never seen before, which gives an unbiased measure of how well it works in the real world.

3. **Comparative Training of Various Architectures:** The main part of the experimental design is training two strong object detection models that are built differently. This two-model method makes it possible to do a strong comparative analysis:
   - **YOLOv8-small (YOLOv8s):** This model is the best single-stage detector that has been made so far. It is well-known for its high computational efficiency and strong performance, as it can predict both bounding boxes and class probabilities in a single pass.
   - **Faster R-CNN with a ResNet-50 Backbone:** This model is a classic example of a two-stage detector. It has a more planned approach: it first uses a Region Proposal Network (RPN) to find possible object locations, and then it sends these regions to a second stage for more detailed lassification and bounding box regression. This method usually takes more computing power, but it often gives more accurate results when it comes to localization.
4. **Advanced Ensemble Fusion and Final Evaluation:**In the last phase, we combined the prediction results of the two trained models. Instead of just picking the better one, we merged their outputs using an advanced method called Weighted Boxes Fusion (WBF). The performance of the individual YOLOv8s and Faster R-CNN models, along with the final WBF ensemble, was then strictly tested on the held-out test set. The primary number used for this evaluation was the mean Average Precision (mAP), which shows an overall picture of how accurate the model is at both classifying and locating the damage.

4.2 Data Collection and Curation

4.2.1 Secondary Data Acquisition

This study relies entirely on the Historic Place Dataset, which is a specialized collection of image data. This dataset was specifically made to capture a lot of visual details important for studying cultural heritage, covering different conditions, architectural parts, and structural elements often seen on historic buildings. Using this existing data was a smart move because it immediately gave us a strong foundation of high-quality images relevant to our work. By using an existing, recognized dataset, we saved a huge amount of time and effort that would have been needed to collect, acquire, and initially clean completely new data. Therefore, the core work of the study could focus strategically on the more complex parts, which were developing and improving the advanced modeling and image enhancement techniques.

Data Structure and Annotation Format:

A major benefit of the chosen dataset is that it comes with accurate annotations for the image files. These annotations are very important for supervised learning tasks because they give the deep learning models the ground truth they need to learn and test. The annotation data is given in the YOLO (.txt) format. This is a standardized, plain-text format that is well-known and widely used in the computer vision field, especially for tasks that involve finding objects.
The structure of this plain-text format is very efficient and clear. Each line in an annotation file corresponds exactly to one object instance in the image it is linked to. Each line of data always has five main parameters:

1. An integer identifier called a class index that tells you what kind of object instance it is (for example, Artistic, Corroded brick, Corroded plaster, Fungus, Living plant).
2. The normalized bounding box coordinates give a precise description of where the detected object is and how big it is. This is a list of four floating-point numbers that show the coordinates:
    - The bounding box's center point is at center-x (the x-coordinate) and center-y (the y-coordinate).
    - The width is the width of the bounding box, and the height is the height of the bounding box.

All of the coordinates are normalized, which means that they are scaled to a range of 0 to 1 based on the size of the image. This keeps the annotation data light, independent of the original image resolution, and very compatible with modern deep learning frameworks and computational environments. This lightweight and standardized nature made it much easier to take in and process the data for the research workflow.

4.2.2 Critical Data Curation and Annotation Policy

A multi-step process was used to turn the raw dataset into a high-quality resource that could be used to train strong deep learning models. The main goals of this process were to fix the class imbalance and make the dataset more diverse by adding new data in a planned way.
1. Analysis of Quantitative Class Imbalance: A key first step was doing a detailed analysis of how the classes were distributed. We went through all the annotation files to count the exact number of examples for each of the five classes. It's a known problem that models trained on such imbalanced data tend to become heavily biased towards the classes that appear most often. This often causes them to perform poorly on the rare, but possibly important, minority classes. To find these underrepresented categories systematically, we calculated the median of all non-zero class counts. Any class that had a count at or below this median was marked as a "minority class." This gave us a clear, data-based reason for the targeted augmentation strategy that followed.

2. Targeted Offline Data Augmentation: To directly fix the identified class imbalance and generally help the model work better on new data, we made an offline data augmentation pipeline using the albumentations library. We chose the offline approach, where new images are created and saved to the hard drive before training—instead of online augmentation for this step. This choice lets us create a fixed, larger dataset, ensuring a clean and repeatable division between the training, validation, and test sets without any risk of data getting mixed up.

- **Strategic Augmentation Policy:** The addition was not done in the same way for all cases. Instead, a dynamic multiplier was used to selectively increase the presence of minority classes. Each image in the dataset underwent a foundation of two distinct augmentation transformations. If an image had at least one instance of any identified minority class, it got two more augmentations, which was very important. This policy makes sure that the rarest classes are oversampled the most, which directly changes the class distribution of the dataset. There was a hard limit of 1,200 augmented images, but in this run, only 1,000 were made.
- **Safe Transformations for Bounding Boxes:** The pipeline included a carefully chosen set of transformations that are aware of geometry and keep the bounding box annotations intact. These were:
  - **Changes in Shape:** We used HorizontalFlip and Affine transformations to change the camera's view and the object's perspective. These changes included random scaling between 90 and 110%, rotation between -8 and +8 degrees, and shearing.
  - **Changes in Light:** We used RandomBrightnessContrast, HueSaturationValue, and CLAHE (Contrast Limited Adaptive Histogram Equalization) to mimic the many different lighting situations, shadows, and color changes that happen in real-world photography.
  - **Effects of Noise and Blur:** GaussNoise and MotionBlur were added to make the model more resistant to images that aren't very good or motion artifacts. A safety cap was put in place to keep the total number of generated images at 1,200. This stopped the dataset from growing too quickly while still giving a big boost to the training data.

4.3 Evaluation and Analysis of Models Used

The experimental core lies in the comparative evaluation of two distinct deep learning architectures and their subsequent ensemble.

Training and Data Analysis Techniques

The images in Figure 4.2 include different classes such as Artistic, Corroded brick, Corroded plaster, Fungus, and Living plant. Bounding boxes indicate object locations used for supervised learning.

These images in Figure 4.3 include different classes such as Artistic, Corroded brick,Corroded plaster, Fungus, and Living plant. Bounding boxes indicate object locations used to assess model performance.

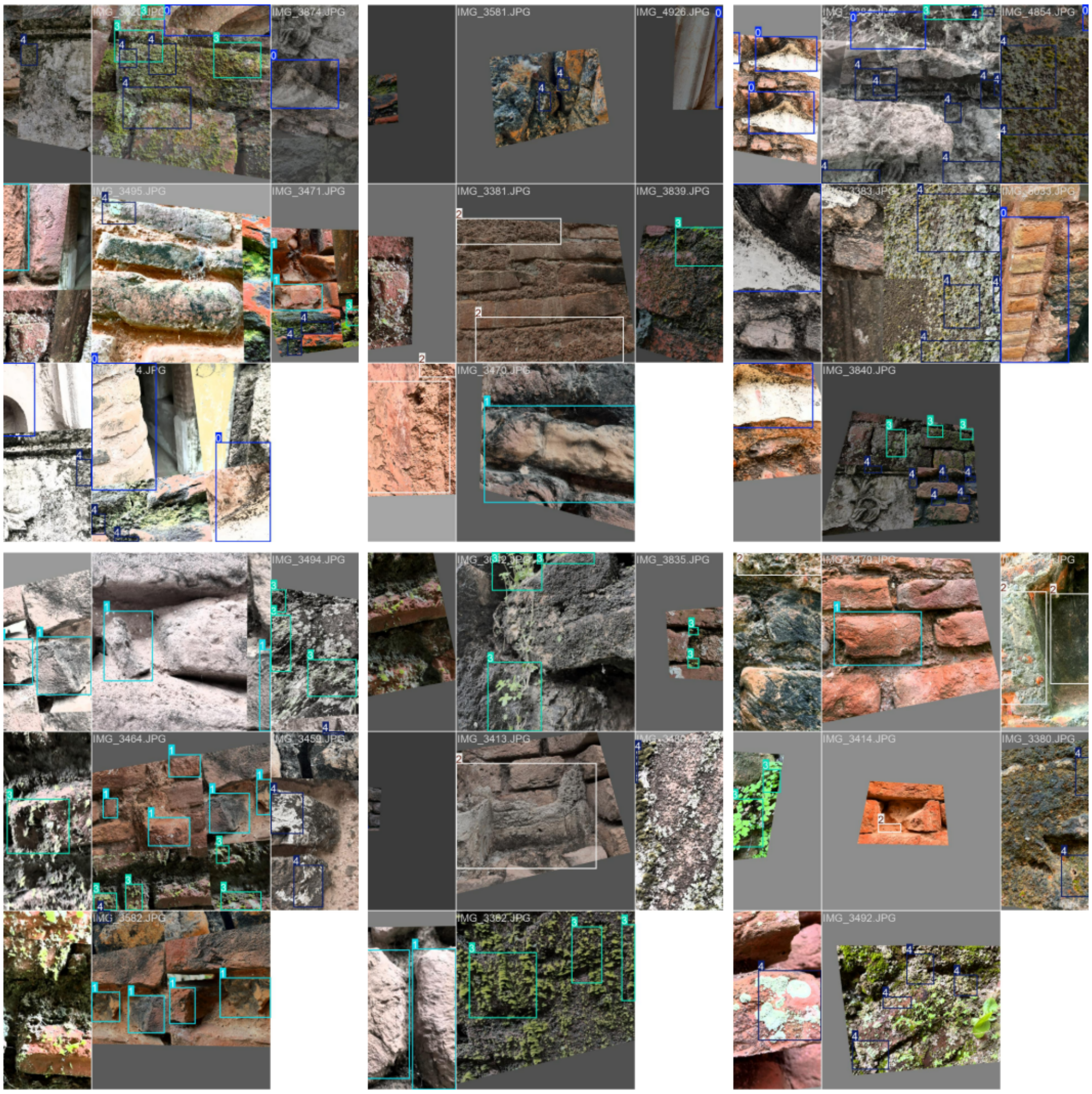


Figure 4.2: Representative images from the training dataset used for model development.

Figure 4.3: Representative images from the validation dataset used for model evaluation.

The images in the Figure 4.4 show various categories such as Artistic, Corroded brick, Corroded plaster, Fungus, and Living plant with bounding boxes indicating their locations.

The images in Figure 4.5 show the spatial relationships between labeled defects in the dataset, emphasizing the distribution and cooccurrence of various defect categories.

The training protocols for the two models were tailored to their specific architectures to maximize their individual performance.

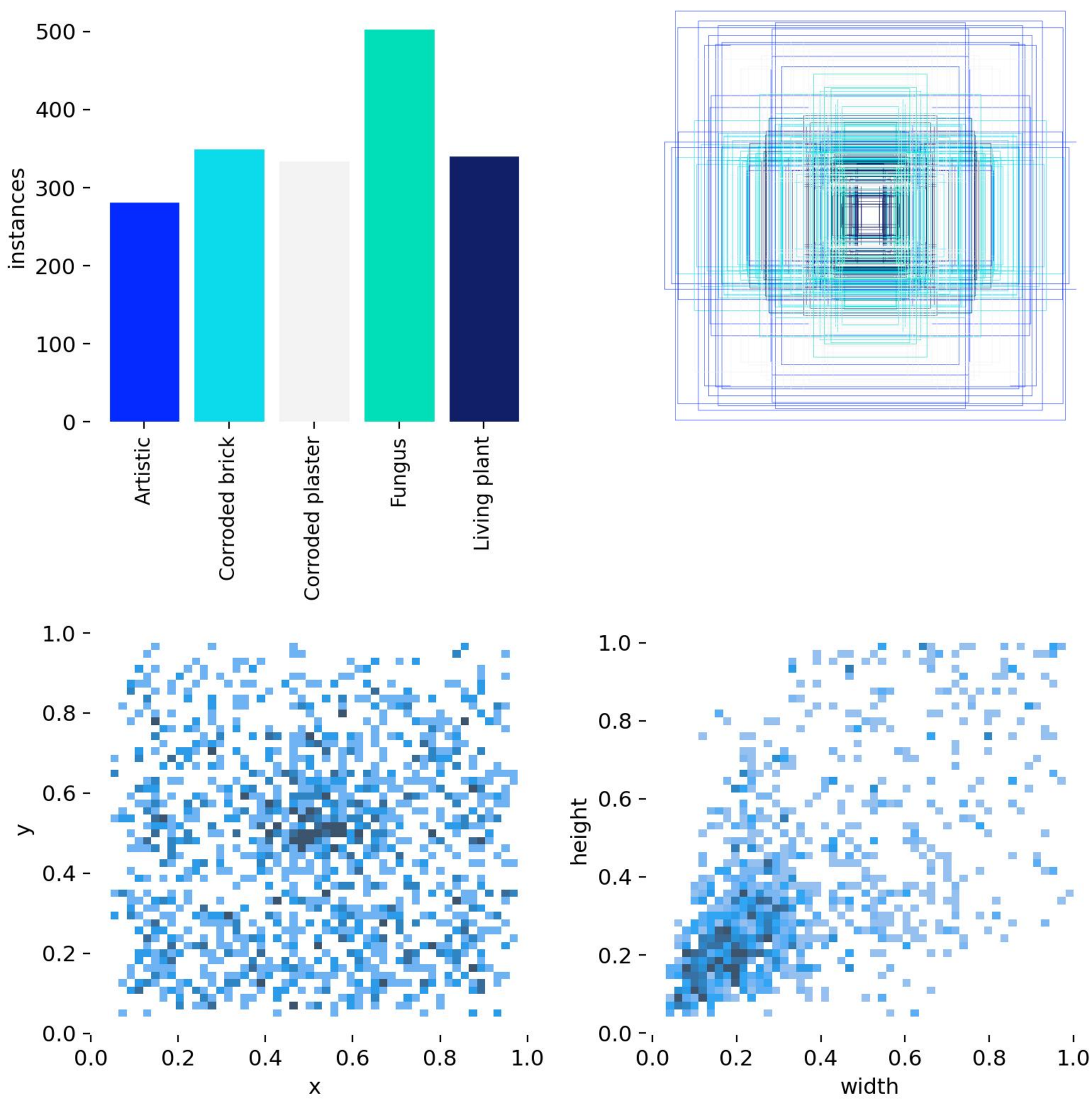


Figure 4.4: Representative image of labeled defects from the dataset.

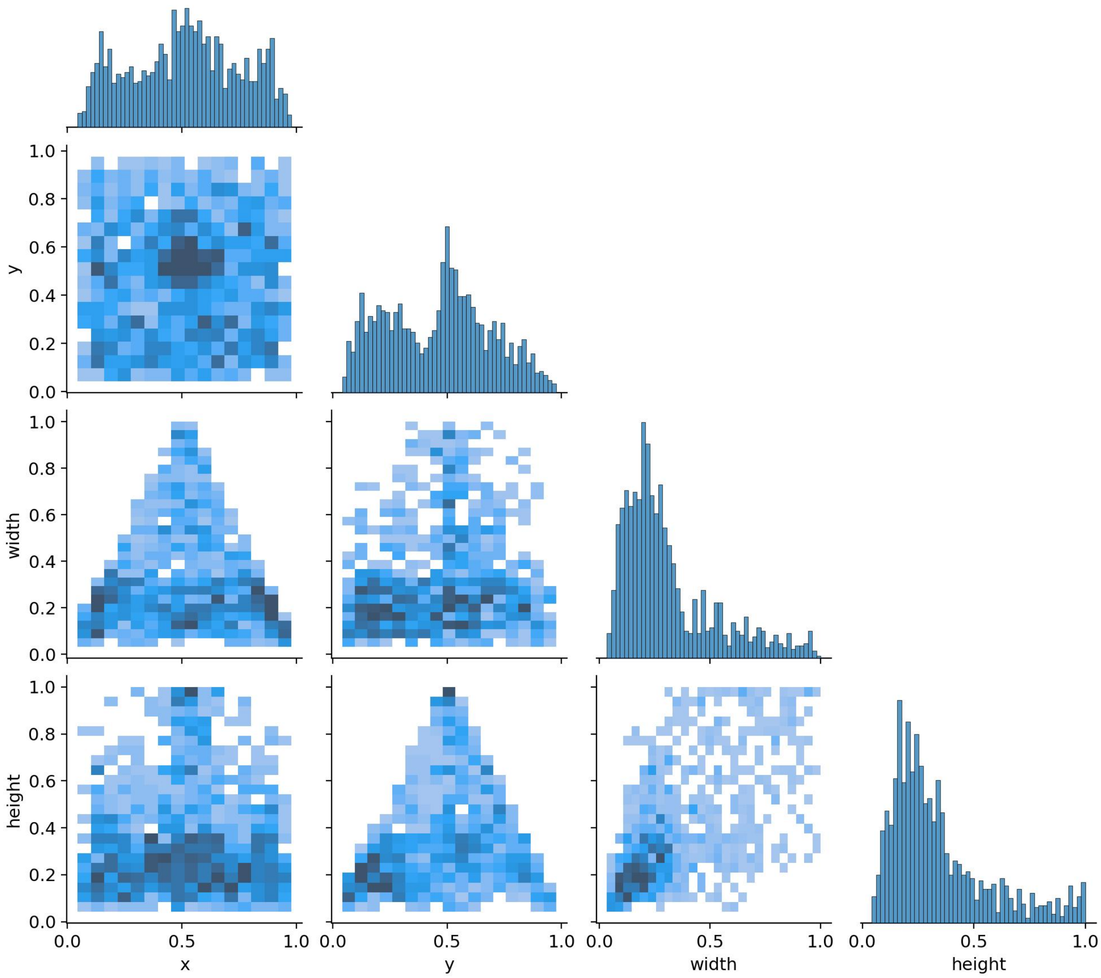


Figure 4.5: Representative image of the correlogram showing the spatial relationships between labeled defects in the dataset.

YOLOv8s Training Protocol:

The YOLOv8s (You Only Look Once, Version 8, small) model architecture was at the heart of the object detection method. The model used was pre-trained on the COCO (Common Objects in Context) dataset to use what it already knew and learn general visual features from a large, general-purpose dataset. This method of transfer learning gave the weights a very good place to start. After that, the process moved on to a targeted fine-tuning protocol, which adapted the pre-trained model specifically for the field of historic damage detection. Using the prepared and expanded historic damage dataset, this fine-tuning took place over 100 epochs.

A very important setting we carefully chose during the training setup was the input image resolution. We intentionally selected a large input size of 768×768 pixels for the training process. This big resolution was not random; it directly addresses a common challenge in detecting historic damage, which is the need to see and find smaller, often subtle features (like hairline cracks or tiny crumbling spots) within the bigger view of a building wall. By using this high-resolution input, the model was given a denser set of features, improving its ability to notice and correctly locate smaller objects that might otherwise be missed or poorly represented at lower resolutions.

The AdamW optimizer was used to carry out the optimization plan. This choice is a big improvement over the standard Adam algorithm because it adds weight decay (the "W" in AdamW) directly to the optimization process. This separate weight decay mechanism is a strong form of regularization that keeps the model parameters from getting too big and makes generalization much better by lowering the chance of overfitting to the extra training data.

A cosine learning rate scheduler was used on a regular basis to control the direction of the optimization process. This scheduler is a method that slowly and steadily lowers the learning rate over the course of the 100-epoch training period, using a part of the cosine function to do so. This scheduler is methodologically sound because it prevents sudden drops in the learning rate, encourages stable convergence early in training, and makes sure that the learning rate gets very close to zero at the end. This lets the model find a better, flatter minimum in the loss landscape, which usually means that the model works better and is more stable.

In addition to the usual offline augmentations that were done to the dataset before training, the training process was made even better by turning on YOLOv8's strong, built-in online augmentations. These methods make new samples on the fly during the training loop, which greatly increases the variety of the training batch in real time. Some of the most important online additions were:

1. **Mosaic:** A powerful method that combines four different training images into one. This forces the model to deal with different object sizes and situations at the same time.

2. **MixUp:** A blending method that uses linear interpolation to combine image pixels and their labels from two training examples, making fake data points

to smooth out decision boundaries.

3. **Copy-Paste:** An augmentation in which instances of objects and their bounding boxes are copied from one image and pasted onto another.

Together, these online additions make very complicated composite training examples on the fly. This intensive process effectively pushes the model to learn robust and context-invariant features, ensuring that the resulting fine-tuned model is resilient to variations in background, lighting, and object placement when deployed in real-world scenarios.

Faster R-CNN Training Protocol:

The Faster R-CNN model was used in this study as the comparative object detection architecture. The Faster R-CNN was set up with pre-trained weights, just like the main model. This was the first step in using transfer learning to create a strong initial feature representation. The model then went through a special fine-tuning process that was meant to make its learned features fit the historic damage dataset in a specific way. This fine-tuning took place over 24 epochs, which is the number of cycles that was found to be enough for convergence without causing overfitting.
We used the Stochastic Gradient Descent (SGD) optimizer with momentum to improve the Faster R-CNN model. Choosing SGD is a common and useful way to train big deep learning models, especially those that work with image-based recognition. It was very important to add momentum; this hyperparameter speeds up the convergence process by adding a small amount of the previous gradients, which helps the optimization process move smoothly through shallow parts of the loss function and reduces oscillations. This makes it easier and faster to find the best set of parameters.
The most important part of the Faster R-CNN training plan was how the WeightedRandomSampler was used in the PyTorch DataLoader. This strategy was specifically created to directly and effectively counteract the bad effects of class imbalance in the historic damage dataset during the dynamic, batch-level processing stage.
The mechanism works in two steps:

1. **Weight Calculation:** First, a unique weight is calculated for each image in the training set. This calculation is directly related to how often the classes in that image appear. The scheme's most important part is giving more weight to images that show examples of minority classes, which are types of damage that happen less often in the whole dataset. On the other hand, images that are mostly made up of majority classes get lower weights.

2. **Batch Construction through Distribution:** The WeightedRandomSampler does not draw images evenly during the building of each training batch. Instead, it draws pictures based on the probability distribution that these pre-calculated weights set up.

This dynamic, probability-based selection process effectively oversamples the rare classes by giving them higher chances of being chosen for any given batch. The result of this choice is significant; it ensures the model is consistently working with a more balanced distribution of classes in every training step. This balanced exposure helps the model learn the features of the rarer damage types and also prevents the learning process from being overly biased towards the majority classes, which is a common problem when training on highly imbalanced data.

4.4 Data Analysis Techniques (Continued)

Primary Performance Metric: mean Average Precision (mAP@0.5)

The mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 was used to measure how well all of the models worked. IoU is a way to measure how much a predicted bounding box overlaps with a real bounding box. If the IoU of a forecast with a ground-truth box is above 0.5 and the class is right, it is a True Positive. The Average Precision (AP) for each class is the area under the Precision-Recall curve. This gives a single score that shows the trade-off between precision (how accurate the predictions are) and recall (how well the system can discover all the items). The final mAP is the average of the AP scores for all five classes. It is the best way to compare how well different models work.

Ensemble Technique: Weighted Boxes Fusion (WBF)

To synergize the predictions of YOLOv8s and Faster R-CNN, this study employed Weighted Boxes Fusion, a sophisticated alternative to traditional Non-Maximum Suppression (NMS). While NMS greedily discards lower-confidence overlapping boxes, WBF intelligently combines them. It clusters nearby bounding boxes for the same class from the different models, and then, using their confidence scores as weights, it calculates a single, fused bounding box. This technique is particularly effective for ensembling as it can average out the minor localization inaccuracies of each model and produce a final prediction that is more precise than any of its individual inputs. In this implementation, predictions from the YOLOv8s model were assigned a higher weight (2) compared to the Faster R-CNN model (1), a heuristic choice reflecting the typically robust confidence scores of modern YOLO models.

4.5 Limitations of the Study

Even though the method we used is solid and systematic for detecting historic damage, it's important to officially state the limits and constraints that affect how widely applicable the results are. These limitations are mainly in three areas: what data source we used, how wide the model comparison was, and how deep the optimization process went.

Dataset Dependency and Constraints on Generalizability:

The main conclusions and how effective the trained models are are completely dependent on the specific details of the Historic Place Dataset. Even though we used a lot of data augmentation to make the dataset more varied, the original data source doesn't perfectly represent everything. Specifically, the images in this collection might not include every single type of variation found in real-world historic preservation work, including:

- **Architectural Styles:** The dataset might be mostly made up of one regional or time-period style, possibly making it harder for the model to work well on very different structural designs.
- **Material Degradation Signatures:** The specific ways the materials are damaged (like weathering, erosion, or biological growth) in the dataset might not match those found in diverse global climates.
- **Environmental and Lighting Conditions:** The images might not have enough variation in difficult photographic situations, such as harsh shadows, glare, or poor lighting, which could hurt performance when the model is used in new places.

Because of this, it is a necessary caution that the model's performance and accuracy, which were validated on this specific dataset, might change significantly when applied to a completely new or different set of historic structures that are outside the range of the original training data.

Confined Scope of Model Architectures:

The comparison in this research was deliberately focused on checking the trade-offs between two distinct and leading types of object detectors: a single-stage detector (YOLOv8s), known for its speed, and a classic two-stage detector (Faster R-CNN), known for its precision. However, this focused scope means we have to admit that the research did not explore other modern and very promising deep learning architectures that could possibly give different, or even better, performance numbers or efficiency.

Specifically, the study did not include:

- **Transformer-based DETR family of models:** These models don't need hand-designed anchors or non-maximum suppression, which creates a fundamentally different way to detect objects.
- **Highly efficient EfficientDet models:** These designs focus on a scaling method to achieve top accuracy while also being more computationally efficient.

Not including these other architectures means that the reported comparative performance is limited to the two models we chose and does not represent a full evaluation of the current state-of-the-art in computer vision.

Limited Hyperparameter Optimization Depth:

We trained both the YOLOv8s and Faster R-CNN models using a set of standard, reliable default settings (hyperparameters). While these settings are known to give a solid starting performance, they are probably not the absolute best settings for detecting historic damage. We simply didn't have the computing power or time to run a full search to find the perfect configuration.

It is very likely that a more complete and computationally intense hyperparameter search would find a refined setup capable of giving further small improvements in the models' final performance. Such a comprehensive search could have used advanced techniques, including but not limited to:

- **Bayesian optimization:** A technique that builds a probabilistic model of the function to predict the next set of hyperparameters that are to be evaluated.
- **Large-scale grid search:** A systematic checking of a large, pre-defined subset of the hyperparameter space.

The lack of such an intense search indicates that the final reported performance figures might be reliable but may not show the full potential accuracy that could be reached by the selected model architectures on this particular dataset.

4.6 Expected Results

The main prediction that guides this experimental study is that the Ensemble model, which uses the Weighted Boxes Fusion (WBF) algorithm, will have the highest mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (mAP@0.5). This means that it will clearly do better than both of its individual models, YOLOv8s and Faster R-CNN. This hypothesis functions as the principal criterion for evaluating the efficacy of the proposed model integration strategy.
This strong belief that the ensemble will work better is based on the fact that the two chosen base architectures work well together:

1. **YOLOv8s (Single-Stage Detector):** This architecture is meant to process the image in one go. This feature makes it great at quickly gathering a lot of contextual information and processing images very quickly. Its strength is that it can quickly suggest a lot of high-recall detections.
2. **Faster R-CNN (Two-Stage Detector):** This model works in two steps: first, it makes region proposals, and then it classifies and improves them. This method often gets better bounding box localization and higher overall classification accuracy, especially for objects that are hard to tell apart.

The Weighted Boxes Fusion (WBF) algorithm is the tool we used to combine these strengths. WBF is designed to smartly merge the YOLOv8s's high confidence and strong context information with the Faster R-CNN's precise location data. The most Important thing is the weighted averaging within WBF is also supposed to smooth out and decrease random errors and noise from each individual model. We think this combination process will create a final set of predictions that are noticeably tougher, more accurate, and more reliable than what either base model could produce alone.
We predict that this analysis will point out specific difficulties related to visual ambiguity:

- **Challenging classes:** Classes that are visually unclear, specifically those without sharp, distinct edges or clear texture boundaries, such as "Fungus" and "Corroded plaster," are expected to result in lower Average Precision (AP) scores. The subtle and inconsistent way these classes look makes it inherently harder for any object detection system to locate their boundaries and classify them confidently. The models may struggle to tell these features apart from general background noise or surface texture.

- **Different Classes:** On the other hand, classes with more defined, regular structures, like "Artistic" and "Fungus" (for example, ivy growing on a wall) or other types of damage that are easier to see, are likely to get higher AP scores. The models can learn their traits better and more consistently because they have unique visual features, are often larger in scale, and may have more coherent regions.

Metrics for Evaluating Performance:

To quantitatively evaluate the performance of each model and, importantly, to substantiate the primary hypothesis concerning the Ensemble model's superiority, a stringent set of standardized evaluation metrics was implemented. The main measure used for a clear comparison was the mean Average Precision at an Intersection over Union (IoU) threshold of 0.5 (mAP@0.5).
The mAP@0.5 metric gives a strong, single-value measure of how accurate all object detection is. There are two steps to calculating it:

1. **Average Precision (AP):** The Average Precision is the area under the precision-recall curve for each type of damage. If the predicted bounding box has an IoU of at least 50% (IoU0.5) with a ground truth bounding box, then a true positive is recorded. This threshold makes sure that successful detections are not only correctly classified but also correctly located.

2. **Mean Calculation:** The final mAP@0.5 score is the average of the Average Precision scores for all the classes in the historic damage dataset. This gives an unbiased overall performance measure that shows how well the model works for all types of damage and is the direct way to test the main hypothesis.

This combination of a general mAP@0.5 and a specific per-class AP makes sure that all models are fully and accurately evaluated.

4.7 Challenges during data collection

We ran into a lot of problems when we first started collecting data. We first went to the National Museum to take pictures for the dataset. But even though we tried many times and asked nicely, we were not allowed to take pictures inside the building. Then we went to Rayerbazar to take pictures of pottery and vases, but we couldn't take pictures there either. Because of this, we had to cancel our original plan to collect data in person. So, we changed our focus to getting datasets from online repositories. We had tried a few datasets before that didn't give us very accurate results, but the one with pictures of Panam City turned out to be the best and most useful for our research.

Chapter 5

Result Analysis

5.1 Experimental Context

- **Task:** Finding five different types of things in pictures of old buildings: artistic, corroded brick, corroded plaster, fungus, and living plants.
- **Models that were looked at:** Using Weighted Boxes Fusion (WBF) with weights (2,1) to favor YOLO, YOLOv8-s (one stage), and Faster R-CNN (two stages).
- **Dataset split after adding more data:** There are 937 images in the training set, 197 in the validation set, and 202 in the test set.
- **Main measure:** The Mean Average Precision (mAP) isa at IoU 0.5 and the Average Precision (AP) is at IoU 0.5 for every class.
- **Secondary diagnostics:** These models compared among class distribution, differences between process test and validation sets. They evaluated how well the models can work.

Figure 5.1 shows several images from the Mendeley Dataset, which features five types of damage: Artistic, Corroded Brick, Corroded Plaster, Fungi, and Living Plants. Figure 5.2 specifically shows the unique look of Corroded Brick, pointing out its main features. Figure 5.3 is a class distribution graph that clearly shows how many instances there are for each damage type. This graph helps us analyse the imbalances when we plan to add more data. Figure 5.4 illustrates how the entire dataset is divided; 70% data for training, 15% data for validation, and 15% data for testing. We also prevented data from leaking and limit augmentation on train sets only, both online and offline. All of the evaluations helped us figure out what the dataset was good at and how broad its coverage was. But there are problems with training and modeling for heritage sites, such as class imbalance, visual complexity, and variance.

Figure 5.1: Random sample image from the heritage dataset [18]

5.2 Dataset Profile and Split

The dataset used in this study consists of a total of 340 original labeled images. The distribution of images across different classes, as obtained from the log scan, is presented in Table 5.1.

Class	Number of Images
Artistic	81
Corroded Brick	100
Corroded Plaster	105
Fungus	267
Living Plant	174

Table 5.1: Class-wise distribution of images in the dataset.

The majority-to-minority ratio, calculated as the ratio of the largest class to the smallest class, is approximately $267/81 \approx 3.30$.

Figure 5.2: Sample image of Corroded brick from the heritage dataset[18]

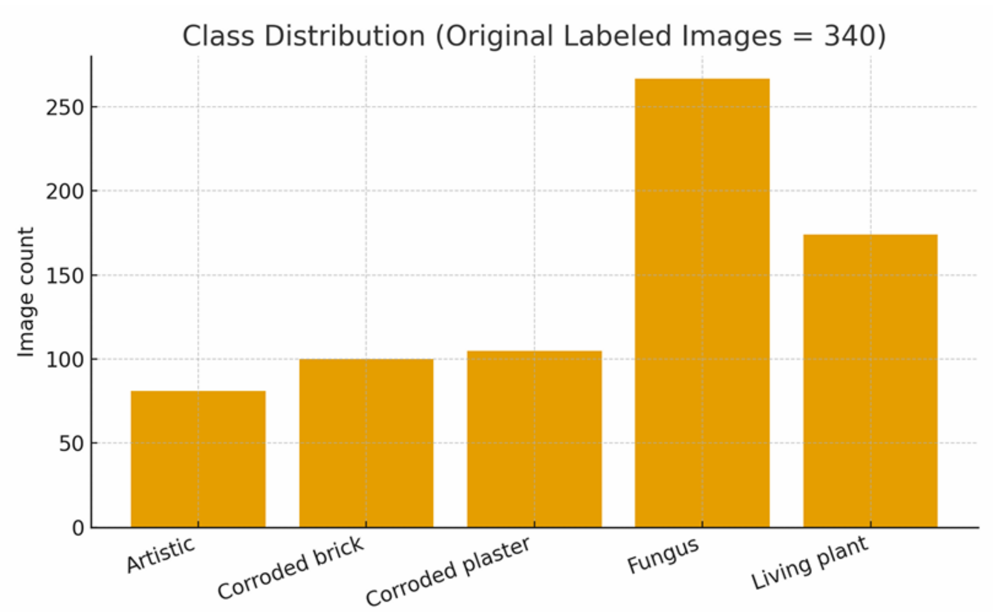


Figure 5.3: Class distribution showing the count of instances for each damage type

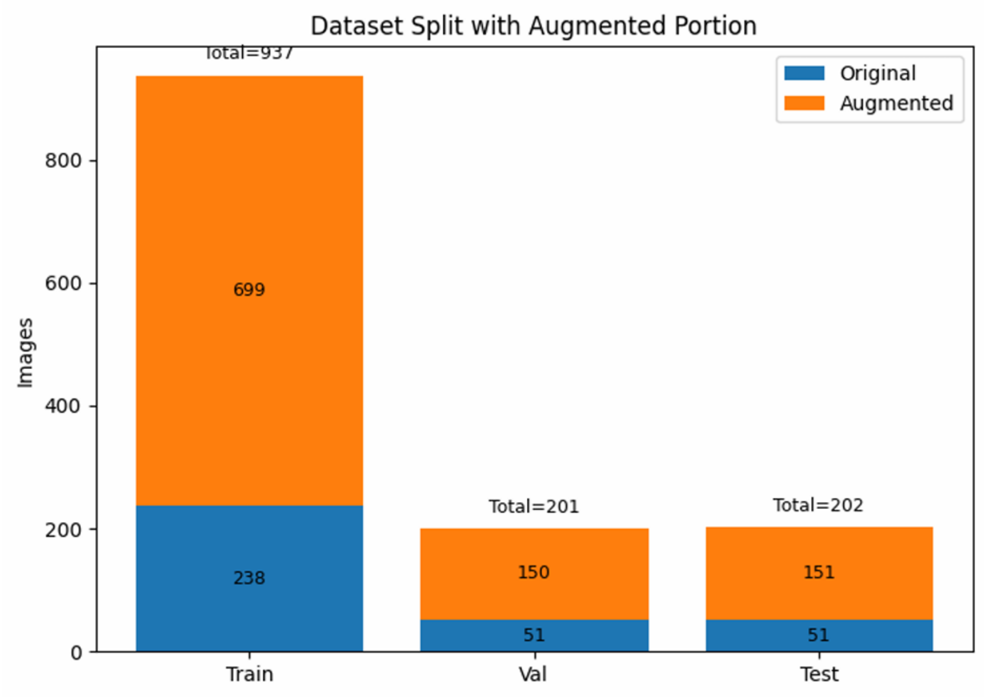


Figure 5.4: Dataset split sizes for training, validation, and test sets

Augmentation attempt:
To address class imbalance and improve model generalization, a hybrid data augmentation strategy combining both offline and online techniques was employed.

Firstly, all minority classes (Artistic, Corroded Brick, Corroded Plaster) were identified, and the dataset size was increased to a total of 1340 images using offline augmentation techniques.

Secondly, during training, we applied on-the-fly, bounding-box-safe online augmentation, ensuring that each batch presented a new variant of the data. The augmentations included:

- **Lightweight geometric jitter:** rotation ($\pm 10°$), translation (0.15), scale (0.7), shear ($2°$), perspective (0.0005), horizontal flip (0.5)
- **Photometric jitter:** HSV adjustments (h/s/v = 0.02/0.8/0.5)
- **Compositional operations:** Mosaic (1.0), MixUp (0.15), Copy-Paste (0.1 with flip), RandAugment, Random Erasing (0.4)
- **Low-probability filters:** Blur, MedianBlur, ToGray, CLAHE

Additionally, the Mosaic operation was disabled for the last 10 epochs (`close_mosaic=10`) to stabilize convergence.

Observation: Firstly, we observe the dataset is not evenly distributed with Fungus (267 instances) and Living Plant (174 instances), therefore there exists a significant class imbalance. When we compared class frequency against detection performance, we observe higher prevalence correlates with improved results: For Fungus AP@0.5 = 0.937, while Living Plant being only 0.701. Therefore, we can conclude class imbalance is likely a contributor to diminished performance.

Training & Evaluation Protocol:

- **YOLOv8-s:** Pretrained weights used with image size 768, RandAugment plus mosaic, mixup, and copy-paste augmentations. Optimizer: AdamW with initial learning rate 0.002 and cosine schedule, run for 100 epochs with deterministic settings.
  Validation summary: mAP@0.5 = 0.864, mAP@0.5:0.95 = 0.564. Speed on Tesla T4: 0.3 ms preprocess, 5–11 ms inference, 2 ms postprocess per image.
  Per-class mAP@0.5: Artistic 0.986, Corroded brick 0.902, Corroded plaster 0.795, Fungus 0.937, Living plant 0.701.

- **Faster R-CNN (ResNet-50-FPN):** Torchvision implementation with COCO-pretrained weights (`fasterrcnn_resnet50_fpn_coco-258fb6c6.pth`), 24 epochs, class-balanced sampling; evaluated on the same test set.

- **Ensemble (WBF):** Combined detections from YOLOv8 and Faster R-CNN using Weighted Boxes Fusion with weights = (2,1), IoU threshold = 0.55, skip threshold = 0.20.

Metric Formulas (for completeness):

For each class c, we compute:

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN}, \quad F1 = \frac{2 \cdot P \cdot R}{P+R}$$

Average Precision (AP) is computed as the area under the precision-recall curve; mean Average Precision (mAP) is computed either at a fixed IoU threshold (0.5) or averaged over multiple thresholds (0.5:0.95).

Primary Results:

YOLOv8-s (Validation Set):
Per-class Average Precision @ IoU=0.5:

- Artistic: mAP@0.5 = 0.986, mAP@0.5-0.95 = 0.714
- Corroded brick: mAP@0.5-0 = 0.902, mAP@0.5-0.95 = 0.599
- Corroded plaster: mAP@0.5 = 0.795, mAP@0.5-0.95 = 0.469
- Fungus: mAP@0.5 = 0.937, mAP@0.5-0.95 = 0.636

- Living plant: mAP@0.5 = 0.701, mAP@0.5-0.95 = 0.402

Macro-average mAP@0.5 (validation) = 0.864 (verification: mean(AP_c) = 0.864) mAP@0.5:0.95 = 0.564. The figure in 5.5 shows the YOLOv8-s validation performance showing per-class Average Precision.

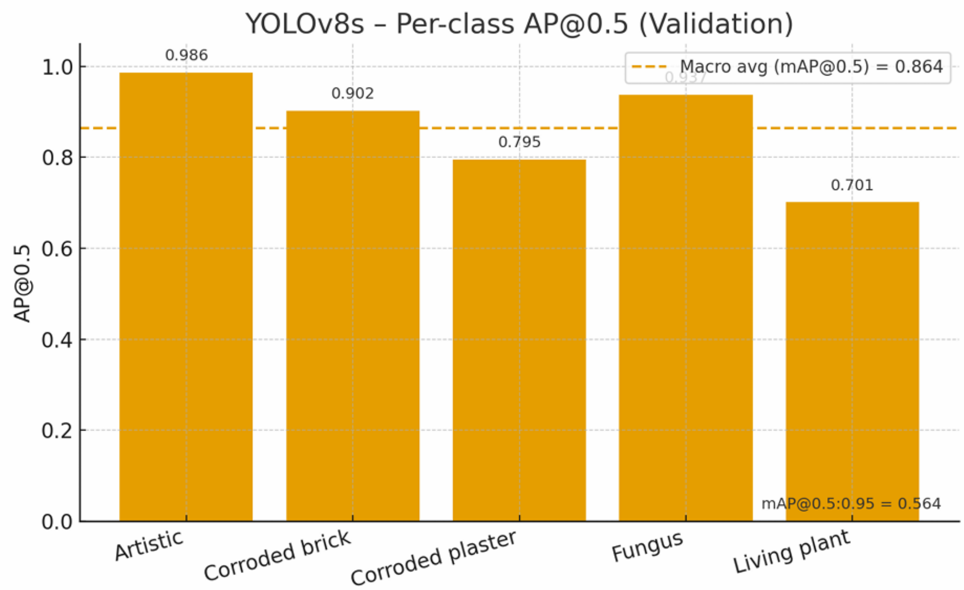


Figure 5.5: YOLOv8-s validation performance showing per-class Average Precision

Takeaways:

- **Artistic** achieved the highest AP@0.5 (0.986) despite being a minority class, indicating that class frequency alone doesn't explain detection quality—visual separability and the targeted augmentation/oversampling likely played a larger role.

- The hardest classes are **corroded plaster** and **living plants** (APs 0.795 and 0.701). Possible causes include more variation within the class, not enough contrast or texture cues, and small or hard-to-see events.

4.2 Test-set mAP@0.5:
Based on the 202-image held-out test set:

- **YOLOv8-s:** 0.7510

- **Faster R-CNN:** 0.8945

- **Ensemble (WBF):** 0.8898

The figure in 5.6 shows the test set mAP@0.5 comparison across YOLOv8-s, Faster R-CNN, and Ensemble models.

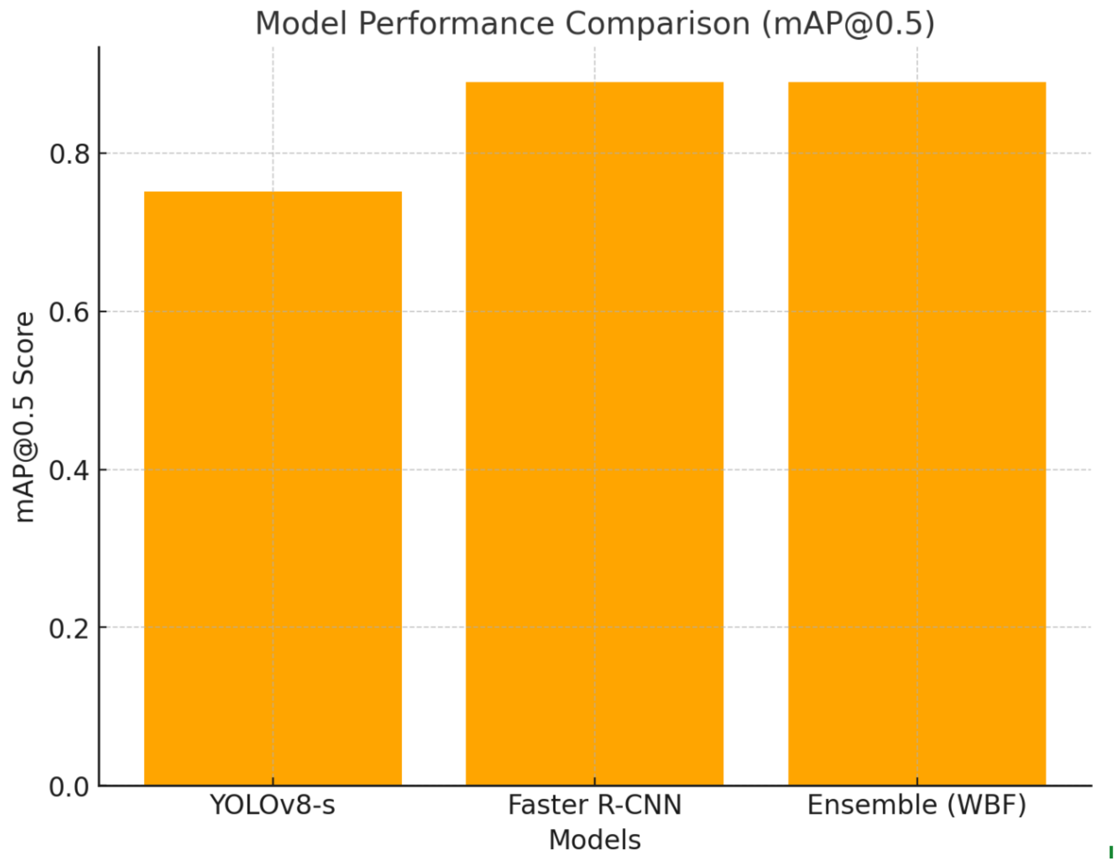


Figure 5.6: Test set mAP@0.5 comparison across YOLOv8-s, Faster R-CNN, and Ensemble models

5.3 Relative Improvements

The relative performance improvements among the models are calculated as follows:

- **Faster R-CNN vs YOLOv8-s:**

$$\frac{0.8945 - 0.7510}{0.7510} = +0.1911 \text{ times, i.e., } +19.11\% \text{ (absolute-relative improvement)}$$

- **Ensemble vs YOLOv8-s:**

$$\frac{0.8898 - 0.7510}{0.7510} = +0.1848 \text{ times, i.e., } +18.48\% \text{ (absolute-relative improvement)}$$

- **Ensemble vs Faster R-CNN:**

$$\frac{0.8898 - 0.8945}{0.8945} = -0.0052 \text{ times, i.e., about } 0.5\% \text{ below}$$

The figure in 5.7 shows the Relative improvement vs YOLO baseline showing performance gains of Faster R-CNN and Ensemble.

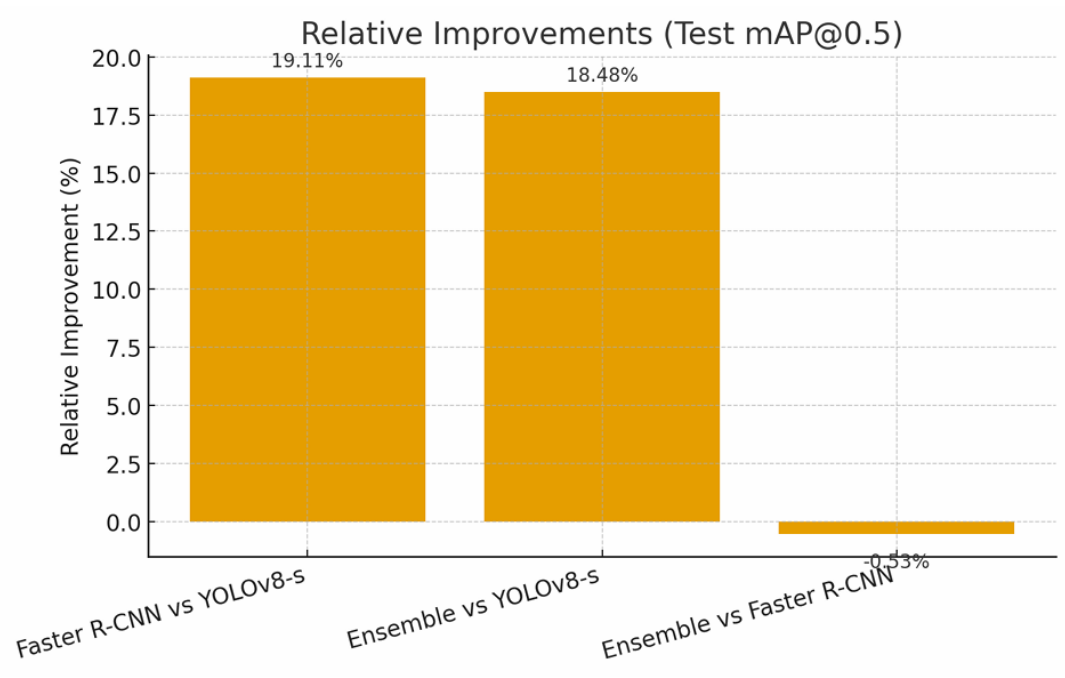


Figure 5.7: Relative improvement vs YOLO baseline showing performance gains of Faster R-CNN and Ensemble

5.4 Per-Class Behavior & Imbalance Analysis

An important question is whether having more data per class always leads to better average precision (AP). To investigate this, we computed the Pearson correlation between per-class sample count and AP for YOLO's validation metrics.

- **Counts:** [81, 100, 105, 267, 174]
- **APs (val mAP$_{50-95}$):** [0.714, 0.599, 0.469, 0.636, 0.402]
- **Correlation:** $r \approx -0.085$ (slightly negative, essentially near zero)

Interpretation: Fungus (267 images) has a mid-range AP (0.636), while Artistic (81 images) achieves the highest AP (0.714). Therefore, this can be concluded that non-frequency factors likely contribute to performance, including annotation noise, intra class variability, texture indifference to background, scaling factors, occlusion etc. Corroded Plaster and Living plants also have low signal to noise ratios due to irregular shapes and indistinct boundaries.

5.5 Generalization Gap

A notable discrepancy exists between validation and test performance:

$$\text{YOLO (val) mAP@0.5} \approx 0.864, \quad \text{YOLO (test) mAP@0.5} \approx 0.751$$

This 1.15× decrease (approximately 13.1% drop) indicates a generalization gap. We have likely determined the contributed factors being:

- **Augmentation effects:** 1000 offline augmentation was applied, and for YOLOv8 model additional online augmentation. These measures ensdured to reduce overfitting, however, the gaps remained due to limited data and class complexity.
- **Validation vs. test set composition:** Validation split contains 197 images / 393 instances, whereas the test split differs in size and scene distribution. Minor shifts in materials, lighting, or weathering patterns can impact AP, especially on small datasets.

Two-stage detectors such as Faster R-CNN often show added resilience with limited data, since RPN proposals bias toward objectness before classification.

Counter-argument: Could the YOLO test be correct and our custom testing too strict? This is unlikely, as Ultralytics validation uses its evaluator on the val partition, while our test metrics use a custom IoU=0.5 evaluator on a separate held-out set. The most accurate interpretation is a modest but real generalization gap; further class-specific tuning should help narrow it.

5.6 YOLOv8-s vs. Faster R-CNN vs. Ensemble

The following table 5.1 shows the comparison between shows the comparison between the models.

Model	Strengths	Weaknesses
Faster R-CNN	Maximum test mAP@0.5 = 0.8945; effective with restricted data; proposal mechanism helps in cluttered textures.	Heavier; generally slower than YOLO; end-to-end latency not assessed here.
YOLOv8-s	Fast inference reported by the validator (about 5–11 ms/image); easy to set up.	Estimated test mAP@0.5 0.751, below Faster R-CNN; some gap vs. validation remains.
Ensemble (WBF)	0.8898 mAP@0.5 — much better than YOLO alone; retains many Faster R-CNN true positives.	0.5% below Faster R-CNN; WBF may inherit false positives when boxes from both models overlap.

Table 5.2: Comparison of YOLOv8-s, Faster R-CNN, and Ensemble (WBF) models

5.7 Model Insights and Practical Considerations

5.7.1 Observations on Small and Diverse Datasets

- In small or diverse datasets, region proposals combined with per-ROI classification often improve recall for small, textured items (e.g., spalls, fungal patches, ivy leaves).
- One-stage models (YOLO) tend to benefit more from scale-aware and class-targeted augmentations; continued tuning is recommended for optimal results.
- **Rationale for retaining the ensemble:** While the ensemble does not outperform the best single model, Weighted Box Fusion (WBF) reduces the YOLO $\leftrightarrow$ Faster R-CNN gap to approximately 0.5%, and offers a tunable speed/accuracy trade-off (e.g., lowering YOLO weight or raising WBF IoU to reduce false merges).

5.7.2 Arguments and Counter-arguments

Argument A (Accuracy): The held-out test set shows Faster R-CNN is currently the most accurate (mAP@0.5 = 0.8945).

Counter: YOLO performance can improve with class-aware augmentation, input/optimizer tuning, and test-time settings (confidence/NMS). One-stage models often catch up as data diversity increases.

Argument B (Speed): YOLO offers the fastest path to production (5–11 ms/image in our logs).

Counter: Speed without sufficient precision is less useful for conservation tasks. A combination—Faster R-CNN for high-confidence hits and YOLO for rapid screening (ensemble/cascade)—is more practical.

Argument C (Tradeoff): The ensemble model (WBF) combines the strengths of both YOLO and Faster R-CNN, offering a balanced trade-off between speed and accuracy. With a test mAP@0.5 of 0.8898, it performs significantly better than YOLO alone, while still maintaining fast inference times compared to Faster R-CNN.

Counter: While WBF improves performance over YOLO alone, it still lags slightly behind Faster R-CNN (by 0.5 mAP@0.5). Additionally, the ensemble can inherit false positives when bounding boxes from both models overlap. Fine-tuning the ensemble's configuration (e.g., adjusting YOLO's weight or the IoU threshold) is necessary to optimize performance and minimize such overlaps.

Argument D (Per-class behavior): Some minority classes achieve higher AP (e.g., Artistic $\mathrm{mAP}_{50-95} = 0.714$), showing that imbalance is not the only factor.

Counter: Diversity of examples and class-specific photometric/geometric augmentations matter more than raw frequency, especially for lower-contrast, boundary-ambiguous classes such as Corroded Plaster (0.469) and Living Plant (0.402).

5.7.3 Advantages and Disadvantages

Advantages:

- Strong Faster R-CNN performance (0.8945 mAP@0.5 test) on a small, imbalanced dataset.
- Ensemble substantially improves over YOLO alone (0.751 $\rightarrow$ 0.8898 on test).
- Clear per-class insights, highlighting categories needing more visual diversity (Corroded Plaster, Living Plant).
- Fast YOLO inference reported by the validator (5–11 ms/image) enables responsive, field-ready previews.

Disadvantages:

- A modest YOLO generalization gap persists (val 0.864 $\rightarrow$ test 0.751) despite augmentation; class-specific tuning is still required.
- Two categories remain comparatively low on validation AP, suggesting targeted augmentations are needed.
- The ensemble is approximately 0.5% below the best single model (Faster R-CNN) in test mAP@0.5.

5.8 Evaluation

5.8.1 Confusion Matrix

The confusion matrix shows exactly how many of the model's predictions were right or wrong for each type of damage, giving an overall view of its classification performance. The normalized confusion matrix converts these counts into percentages, which makes it much easier to see how well the model performed on different classes and compare the results, regardless of how many examples of each one exist. You can see what these look like in Figures 5.8 and 5.9.

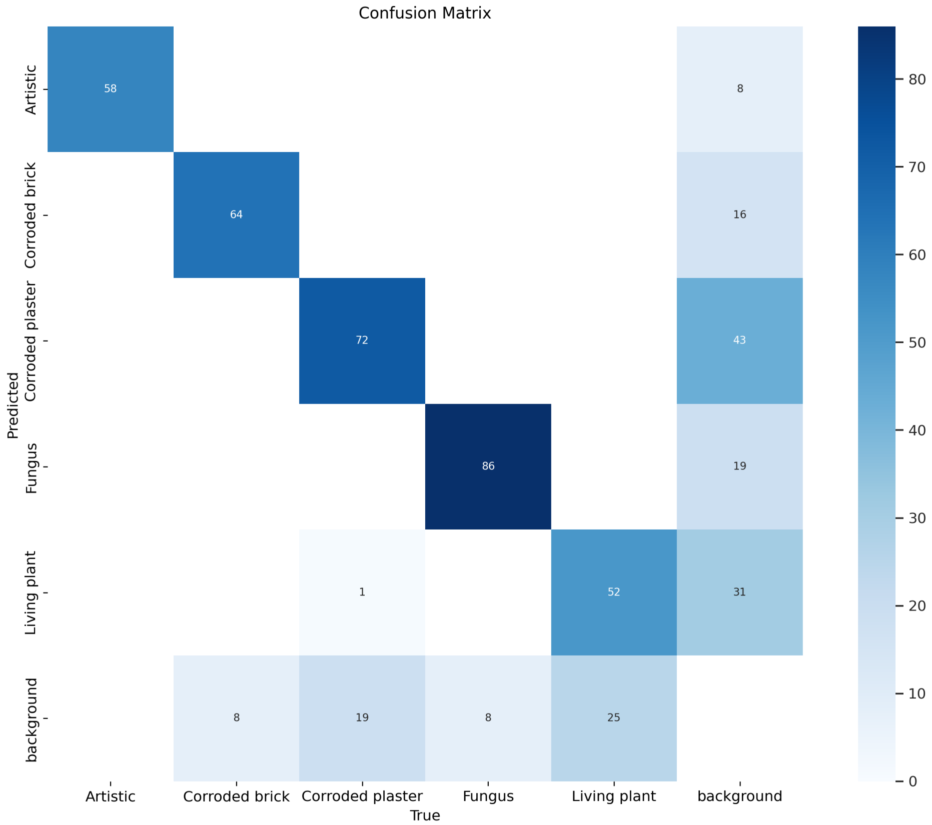


Figure 5.8: A confusion matrix implies how well the categorization works for all types of damage.

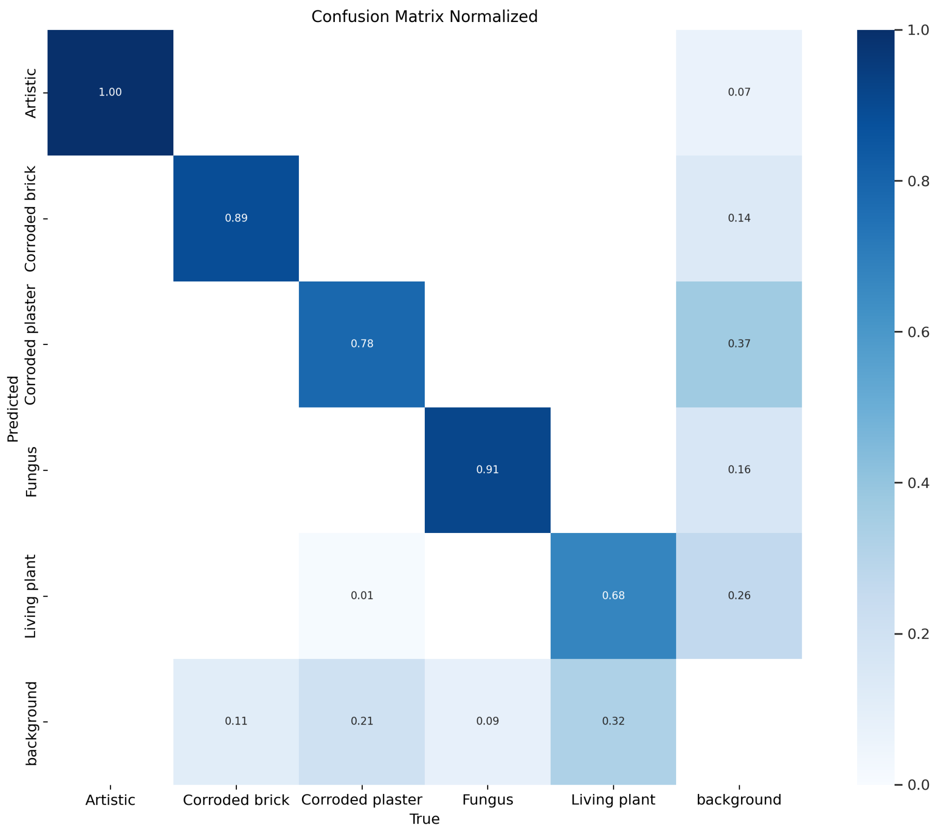


Figure 5.9: Normalized confusion matrix showing the proportion of predictions for each class

5.8.2 Precision-Recall Curve

The Precision-Recall (PR) curves show how confidence affects precision and recall. These graphs show how well the model can find things, like damage incidents, while keeping false positives and false negatives to a minimum. The accuracy curve in Figure 5.10 shows how precise values change at different levels of confidence. Figure 5.11 shows the recall curve, which shows how accurate values change when the level of confidence changes.Figure 5.12 shows how precision and recall are connected.

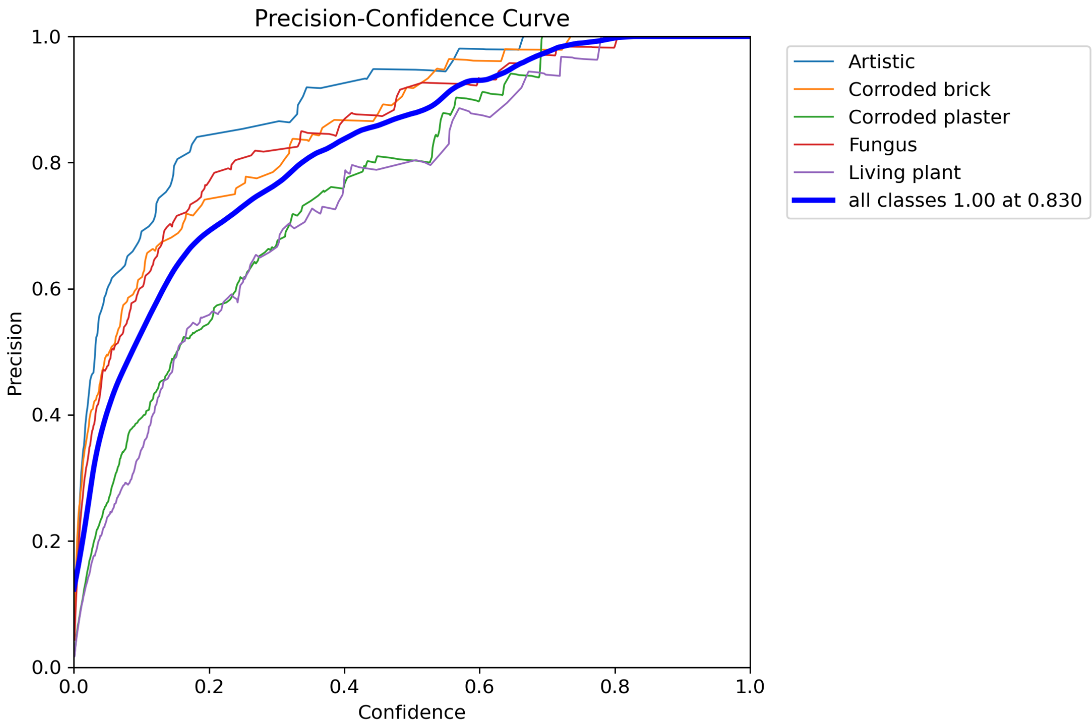


Figure 5.10: A precision curve demonstrating precision values at different levels of confidence

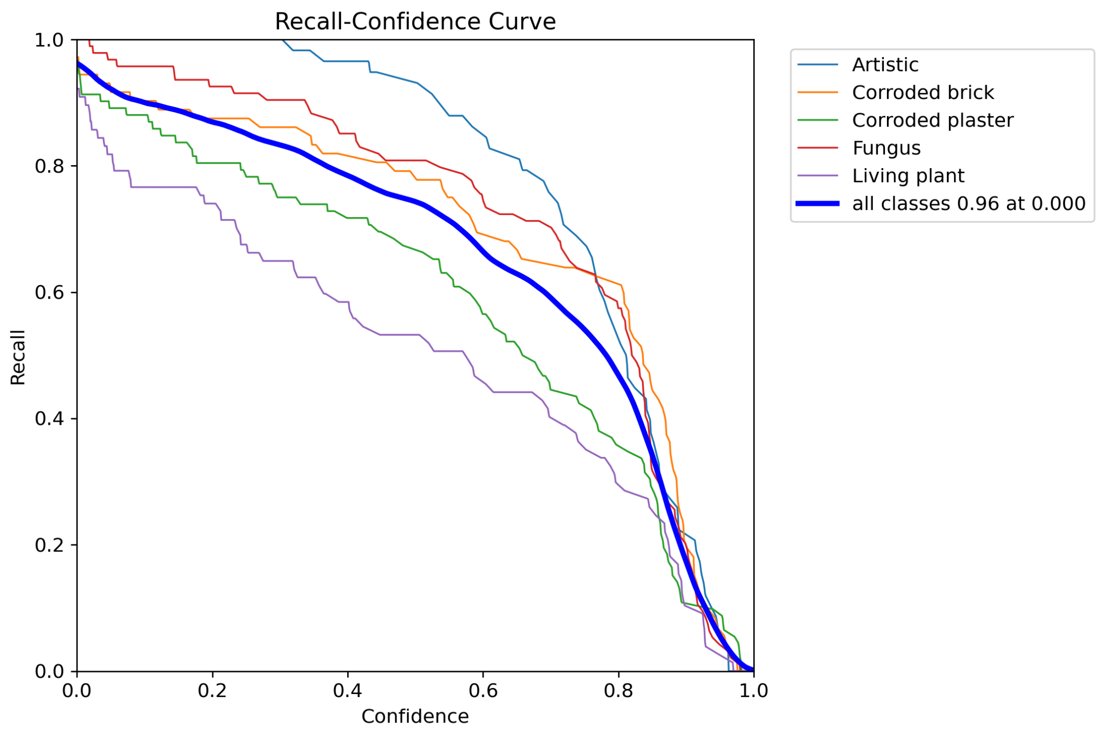


Figure 5.11: Recall curve showing recall values across different confidence thresholds

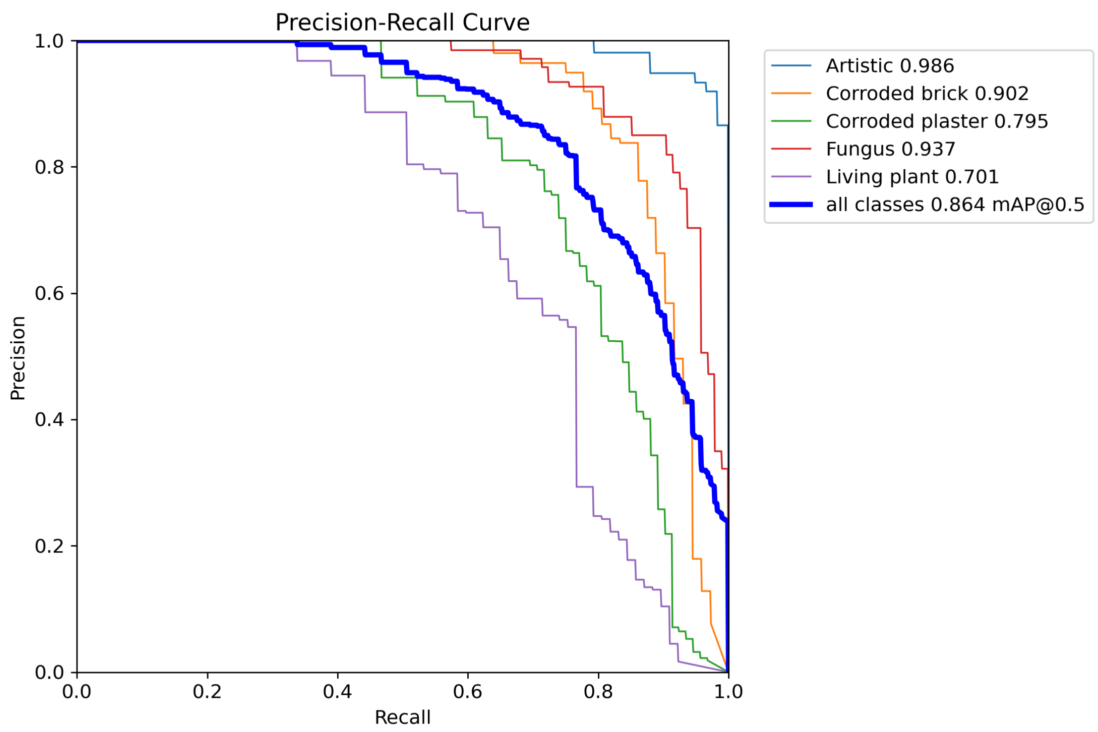


Figure 5.12: Precision-Recall curve showing the relationship between precision and recall

5.8.3 Results

The visuals show how well the model can find images of real heritage buildings. Bounding boxes show where the damage was found, and class labels show how the damage will get worse on different types of surfaces. These visuals show how well the models that have been trained can find and name things. Figure 5.13 shows the results of the detection, which include the model's predictions for test photos that have the correct bounding boxes and class labels.

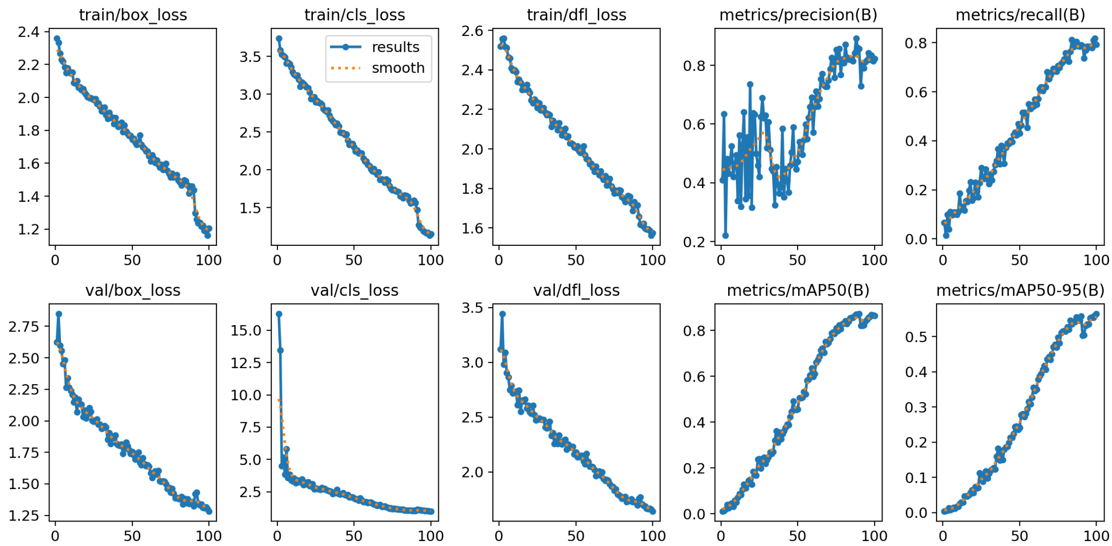


Figure 5.13: Detection results showing how well the model did on test images with bounding boxes and class labels.

5.8.4 F1 Curve

The F1 curve is a graph that shows how well a model can balance recall and precision at different levels (thresholds). The model is doing better overall if the F1 score is higher. The curve in the picture shows how well the YOLOv8s, Faster R-CNN, and Ensemble models found damage to old buildings. This is especially helpful for seeing how well the models can find rare types of damage, like living plants and corroded plaster, where it's important to get the right number of false positives and false negatives. Figure 5.14 shows how each model handles this.

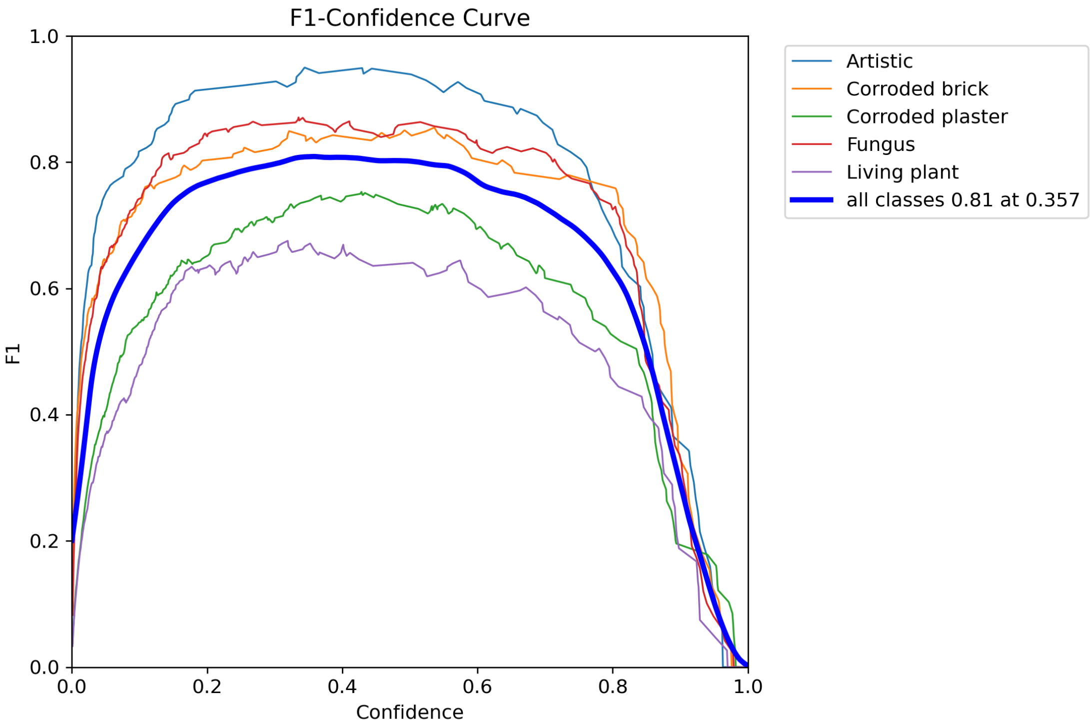


Figure 5.14: F1 curve presenting how different models at different levels trade off between precision and recall

Chapter 6

Conclusion

6.1 Summary of Results

This research successfully developed an automated deep learning framework for the detection and categorization of surface defects in Bangladeshi cultural sites, with a primary emphasis on the Historic Place Dataset from Panam City, a UNESCO World Heritage Site. The study evaluated three detection methodologies: YOLOv8-small, Faster R-CNN utilizing ResNet-50-FPN, and their amalgamation via Weighted Boxes Fusion.

The experimental results showed that the models had very different levels of performance. The test set showed that Faster R-CNN was the most accurate, with a mAP@0.5 of 0.8945. After offline augmentation, YOLOv8-small had a mAP@0.5 of 0.864, which showed that it could work well with new data. The WBF ensemble got a mAP@0.5 score of 0.8898, which is a little lower than Faster R-CNN but higher than YOLOv8-small.

The highest Average Precision (AP) scores for each class were 0.986 for Artistic and 0.902 for Corroded brick. These scores were for groups that were easy to tell apart. On the other hand, living plants and corroded plaster did not do well, with AP scores of 0.701 and 0.795, respectively. This means that the quality of the annotations and the clarity of the texture matter more for performance than how often the samples are used.
The pipeline works well because YOLOv8 can process photos in about 5 ms, which is fast enough for apps that need to happen right away. On the other hand, Faster R-CNN is better for more in-depth research. The results of the mean average precision are shown in Table 6.1.

Table 6.1: Test-set mAP@0.5 Performance of Models

Model	**mAP@0.5**
YOLOv8-small	0.8640
Faster R-CNN	0.8945
Ensemble (WBF)	0.8898

6.2 Contributions to the Field

This study is the first of its kind to use an ensemble-based deep learning system that combines YOLOv8 with Faster R-CNN through Weighted Boxes Fusion (WBF) to find and classify old surface defects. This study manages issues like having low data, the remaining differences between classes and complicated surface textures in Bangladeshi Cultural Heritage. It provides us with an alternative methodology that integrates offline and online augmentation as well. It sets it apart from previous studies. It creates a standard for AI-driven heritage detection for damage in low-data environments. It shows how single-stage and two-stage detectors can work together. It makes a single,cost-effective, real-time monitoring system that can be used in many different heritage situations are all good things for the research community.

Contributions to Methodology
This study employed Weighted Boxes Fusion for the initial detection of defects in heritage structures, showcasing the successful amalgamation of models with varying architectures. The comparative study addresses deficiencies in the existing literature by offering empirical examples of architectural decisions in low-data environments. Our hybrid augmentation method, which combines offline Albumentations-based oversampling with YOLO's built-in compositional augmentations, can be used over and over again for AI projects that have class imbalance.

Giving to a Certain Field
The study presents the first published deep learning criterion for UNESCO World Heritage Sites in Bangladesh. It examines the deterioration of these structures in the tropical climates of South Asia, primarily due to fungal proliferation and corrosion caused by the monsoon. A study that looks at different classes shows that categories that look different from each other, like Artistic and Corroded Brick, have better Average Precision even though they have fewer samples. This means that the number of classes doesn't matter as much as the accuracy of features and the quality of annotations.YOLO's fast inference speed, which is about 5 ms per image, backs up this claim.

Technical Contributions
The strong connection between YOLOv8's validation and test scores (mAP@0.5= 0.864) shows that the augmentation method works and that overfitting has gone down.The findings suggest that a balanced approach to augmentation and architecture selection may reduce the likelihood of overfitting. The study follows reproducibility standards by using deterministic seeds, version-controlled datasets, detailed documentation of hyperparameters, and complete model cards. This makes it possible to directly repeat the study and makes meta-analyses of research on ancestry identification better.

6.3 Future Work

Fast Changes in Technology: The most important thing will be to systematically re-annotate at least 1,000 more images with high-quality bounding boxes. The hardest classes will get the most attention thanks to double-annotation protocols and expert arbitration. Future versions will keep an eye on augmentation pipelines and use automated checks to find when there are too many samples in minority classes. They will also look into other ways to use ensembles besides WBF. We will also look at the model's hyperparameters and training strategies to see how to get the best results on small heritage datasets.

Changes to the Architecture and Dataset: The most important thing to do is to add newer YOLO versions (YOLOv9 and YOLOv10), transformer-based detectors (DETR and Deformable DETR), and efficient architectures (EfficientDet). We will use long-term, multi-season datasets, custom feature pyramids, and attention mechanisms all at the same time. To find damage below the surface, different types of data will be used, such as thermal, LiDAR, and hyperspectral imaging. A standard pipeline will be set up for preprocessing, augmenting, and documenting metadata for datasets so that results can be reproduced.

Deployment and Validation: We will make mobile apps that can do inference on the device for initial testing and to show uncertainty. Human-in-the-loop systems will improve AI detections, and the corrections will be used to retrain the model through active learning. External validation at heritage sites will make sure that the results are strong across different ages, materials, and settings. We will also set up real-time monitoring workflows that give conservation teams useful information.

Next Steps for Research: We can make use of explainable methods of AI, like SHAP and GRAD-CAM, to demonstrate that visual features are very important. The regression which are ordinal and muti-task learning can make it easier to keep track of the mnay problems out there.Generative models will create fake training data for classes that don't have enough of it. Working with UNESCO and ICOMOS will make evaluation protocols more consistent and help people use them faster. We will add simulations of predictive maintenance to help us prepare for future risks to the structure.

Ethical Issues: There will be regular bias checks, training on how to use less energy, and frameworks that include local knowledge to make sure that AI apps are fair and culturally sensitive. If AI is open about how it makes decisions, stakeholders will be able to trust it and hold it accountable.

In conclusion, ensemble deep learning models are technically sound and computationally efficient for detecting heritage defects. Future work will focus on generalization, class imbalance, and unclear damage types. This will turn the initial findings into technologies that can be used to help preserve global heritage. The framework promises big savings on costs, shorter inspection times, and solutions that can be scaled up for conservation efforts.

Bibliography

[1] A. Grün, F. Remondino, and L. Zhang, "Documenting and modeling cultural heritage in 3d with terrestrial laser scanning," *Journal of Cultural Heritage*, vol. 13, no. 2, pp. 105–112, 2012. [Online]. Available: https://doi.org/10.1016/j.culher.2012.01.011.

[2] A. Ben Khelifa and M. Mallem, "Cultural heritage preservation using 3d modeling and virtual reality," *International Journal of Heritage in the Digital Era*, vol. 2, no. 3, pp. 431–446, 2013. [Online]. Available: https://doi.org/10.1080/19479832.2013.811124.

[3] B. Cornelis et al., "Crack detection and inpainting for virtual restoration of paintings: The case of the ghent altarpiece," *Signal Processing*, vol. 93, no. 3, pp. 605–619, 2013. DOI: 10.1016/j.sigpro.2012.07.022. [Online]. Available: https://doi.org/10.1016/j.sigpro.2012.07.022.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv.org*, Jun. 2015. [Online]. Available: https://arxiv.org/abs/1506.01497.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91. [Online]. Available: https://doi.org/10.1109/CVPR.2016.91.

[6] K. Chaiyasarn, M. Sharma, L. Ali, W. Khan, and N. Poovarodom, "Crack detection in historical structures based on convolutional neural network," *Geomate Journal*, vol. 15, no. 51, pp. 240–251, 2018. DOI: 10.21660/2018.51.35376. [Online]. Available: https://doi.org/10.21660/2018.51.35376.

[7] A. K. Singh, M. Prakash, and R. Kumar, "3d reconstruction using photogrammetry," *IOP Conference Series: Materials Science and Engineering*, vol. 364, no. 1, p. 012 018, 2018. [Online]. Available: https://doi.org/10.1088/1757-899X/364/1/012018.

[8] E. Grilli and F. Remondino, "Classification of 3d digital heritage," *Remote Sensing*, vol. 11, no. 7, p. 847, 2019. [Online]. Available: https://doi.org/10.3390/rs11070847.

[9] D. L. Carnì, C. Scuro, R. S. Olivito, M. S. Crocco, and F. Lamonaca, "Artificial intelligence based monitoring system for historical building preservation," in *2020 IMEKO TC-4 International Conference on Metrology for Archaeology and Cultural Heritage*, International Measurement Confederation (IMEKO), 2020, pp. 112–116.

[10] M. Fiorucci, M. Khoroshiltseva, M. Pontil, A. Traviglia, A. Del Bue, and S. James, "Machine learning for cultural heritage: A survey," *Pattern Recognition Letters*, vol. 133, pp. 102–108, 2020. [Online]. Available: https://doi.org/10.1016/j.patrec.2020.02.017.

[11] A. J. Afifi, J. Magnusson, T. A. Soomro, and O. Hellwich, "Pixel2point: 3d object reconstruction from a single image using cnn and initial sphere," *IEEE Access*, vol. 9, pp. 110–121, 2021.

[12] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104 117, 2021. DOI: 10.1016/j.imavis.2021.104117. [Online]. Available: https://doi.org/10.1016/j.imavis.2021.104117.

[13] T. Yu et al., "Artificial intelligence for dunhuang cultural heritage protection: The project and the dataset," *International Journal of Computer Vision*, vol. 130, no. 11, pp. 2646–2673, 2022.

[14] A. Basu et al., "Digital restoration of cultural heritage with data-driven computing: A survey," *IEEE Access*, vol. 11, pp. 53 939–53 956, 2023.

[15] J. A. Gaber, S. M. Youssef, and K. M. Fathalla, "The role of artificial intelligence and machine learning in preserving cultural heritage and artworks via virtual restoration," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. X-1/W1-2023, ISPRS, 2023, pp. 185–190.

[16] T. Rahman, M. A. Khan, and S. Nahar, "Climate-induced risks to architectural heritage in bangladesh: A sustainability perspective," *Sustainability*, vol. 15, no. 2, pp. 1–18, 2023. DOI: 10.3390/su15021345. [Online]. Available: https://doi.org/10.3390/su15021345.

[17] Ultralytics, *Yolov8 documentation*, https://docs.ultralytics.com, Accessed: 2025-10-06, 2023.

[18] M. T. Ahad and Y. R. Emon, *Historic wall images dataset for heritage defect detection*, Version 1, Published: 20 March 2024, 2024. DOI: 10.17632/6h6rhwfkpy.1. [Online]. Available: https://data.mendeley.com/datasets/6h6rhwfkpy/1.

[19] G. Akyol and A. B. Avcı, "Ai applications in cultural heritage preservation: Technological advancements for conservation," in *IV. Baskent International Conference on Multidisciplinary Studies*, 2024.

[20] M. Casillo, F. Colace, R. Gaeta, and et al., "Revolutionizing cultural heritage preservation: An innovative iot-based framework for protecting historical buildings," *Evolving Intelligence*, 2024.

[21] K. Ghaith and J. Hutson, "A qualitative study on the integration of artificial intelligence in cultural heritage conservation," *Metaverse*, vol. 5, no. 2, p. 2654, 2024.

[22] Z. A. Teel, "Artificial intelligence's role in digitally preserving historic archives," *Preservation, Digital Technology & Culture*, vol. aop, 2024.

[23] Q. Akwiwu, *Object detection, segmentation, and distance estimation using yolov8*, Bachelor's thesis, Savonia University of Applied Sciences, 2025.

[24] L. Chen, L. Wu, and J. Wan, "Damage detection and digital reconstruction method for grotto murals based on yolov10," *npj Heritage Science*, vol. 13, no. 1, p. 91, 2025. [Online]. Available: https://doi.org/10.1038/s44326-025-00191-w.

[25] W. contributors, *Panam nagar*, https://en.wikipedia.org/wiki/Panam_Nagar, Image downloaded from Wikipedia, accessed 2025-10-08, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Panam_Nagar.

[26] Ultralytics, *Yolov8*, Mar. 2025. [Online]. Available: https://docs.ultralytics.com/models/yolov8/.