Real-World Evaluation of Automated Defect Detection in Masonry Bridges Using 360° Imagery with Machine Learning

Abstract

Purpose:

This study evaluates different deep learning approaches, CNN, transformer, hybrid, and commercial models, for automated defect detection in UK masonry railway bridges, in both laboratory and real-world settings, using high-resolution 360° imagery.

Design/methodology/approach:

Expert-annotated imagery was categorised into six defect types, with SMOTE oversampling applied to mitigate class imbalance. Four widely used architectures, EfficientNet, Swin Transformer, ConvNeXt, and Azure CustomVision, were benchmarked using compact variants in a two-stage design: laboratory data and real-world evaluation, to assess feasibility and generalisability.

Findings:

All models achieved high performance on laboratory data (0.83 - 0.91 accuracy), demonstrating feasibility in controlled environments. However, when applied to real-world evaluation, accuracies declined to 0.76 - 0.86, with the Swin Transformer showing the greatest robustness (2% drop). This decline was largely attributable to extreme class imbalance (non-defect to defect ratio around 220:1), which caused models to favour the non-defect class. While Vegetation and Loss of Section showed moderate recall, crack detection was less reliable, likely affected by limited samples and textural similarity to other classes. Consequently, overall accuracy masked substantial class-level disparities, and ensemble modelling delivered only marginal improvements under these conditions.

Originality:

This study is the first comprehensive evaluation on masonry railway bridges with 360° imagery, which advances beyond prior laboratory environment by systematically testing generalisability in real-world sceneries, generating new insights into imbalance-driven errors and class-specific detection limits.

Practical implications:

Automated detection can streamline inspections and enhance consistency, as compact models show feasibility. However, reliable deployment requires addressing imbalance, since some defect classes (e.g. cracks) remain unreliable.

Keywords: Automated Defect Detection, 360° Imagery, Machine Learning, Masonry Bridge.

1. Introduction

Masonry bridges have been a vital component of the UK transport network for over a century, with approximately 40% of the nation's bridge stock comprising masonry structures (Majtan *et al.*, 2023). To ensure their continued performance and extend service life, asset owners must undertake routine inspections at prescribed intervals, following established protocols (Washer *et al.*, 2016). Traditional bridge examination techniques involve site visits by experienced engineers, who manually assess the structure's condition and record observations in real time using handwritten notes. Detailed examinations may include tactile inspections of bridge elements to detect issues, such as loose bricks or drummy walls, using examination hammers (Network Rail, 2018). Measuring instruments such as rulers, crack gauges, plumb points and inclinometers provide quantitative data, while photographic evidence complements handwritten records, offering context for condition ratings and justifying any reduced scores due to observed defects or deterioration (Phares *et al.*, 2004). Nonetheless, these conventional procedures are time-consuming and often cause traffic disruptions, road closures and safety hazards (Talebi *et al.*, 2022). Examiners frequently require ladders, mobile elevating work platforms, road–rail vehicles, and scaffolding to access various parts of the bridge (Network Rail, 2018). Dependence on handwritten notes can introduce subjectivity and errors, potentially compromising documentation accuracy (Phares *et al.*, 2004). Moreover, assessors must perform complex calculations to arrive at overall condition ratings, necessitating specialised training (Abdallah *et al.*, 2022). Digital inspection methods, such as 360° imaging, point cloud data, flat images and video, offer significant advantages over traditional techniques in terms of cost, precision and speed, while conforming to established inspection standards (Omer *et al.*, 2021; Wells and Lovelace, 2017). Recent advances in 360° imaging have transformed infrastructure inspection by producing immersive, comprehensive panoramas stitched from multi-lens cameras. Unmanned aerial vehicles (UAVs) equipped with 360° cameras can safely capture hard to access areas of bridges and other tall structures (Chen *et al.*, 2019). Studies demonstrate the efficacy of 360° imaging for monitoring structural assets, detecting cracks and spalling in concrete bridges (Chow *et al.*, 2021), surface damage in heritage buildings (Masciotta *et al.*, 2023) and culvert wall inspections (Meegoda *et al.*, 2019). Furthermore, combining point cloud data with visual imagery enables quantitative defect analysis (Mirzazade *et al.*, 2021). These approaches yield robust, quantitative data that support data-driven decision making in asset management (Wells and Lovelace, 2021).

1.1 Masonry Structure Defects

Common defects on masonry surfaces include cracks, spalling, joint deterioration and vegetation growth. Noy and Douglas (2005) classified masonry defects into categories such as wall bulging and leaning, bonding failures, joint defects, crack development, surface corrosion and defective cavity walls. A technical report by the Michigan Department of Transportation (Zekkos *et al.*, 2020) further identifies defects including spalled masonry, patched masonry, efflorescence, mortar breakdown, masonry displacement and general damage. According to the Asset Data Management Manual from National Highways (National Highways, 2021), defects are categorised as masonry cracking, spalling, scaling, peeling, bulging, missing units, water or wind induced erosion, mortar loss and soft mortar. Network Rail's classification lists defects as: (a) bulging; (b) crack, fracture or ring separation; (c) spalling; (d) joint defects; and (e) loss of section (Sen *et al.*, 2025). Figure 1 illustrates typical image slices of these defects; bulging is seldom visible in 2D images and is therefore excluded.

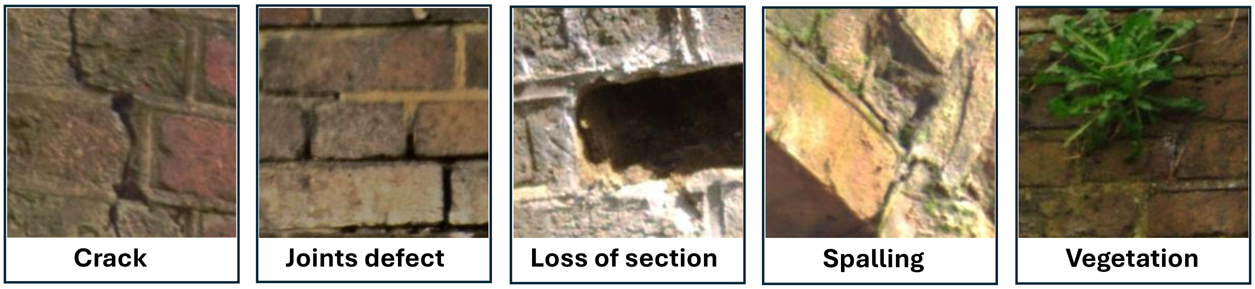


Fig. 1. Examples of masonry defects

1.2 Machine Learning for Defect Detection

Machine learning (ML) has transformed structural inspection by automating defect analysis in 360° imagery (Humpe, 2020). Three key approaches prevail: classification, which distinguishes intact from damaged surfaces (Deng *et al.*, 2021); object detection, which locates defects with bounding boxes (Teng *et al.*, 2022); and segmentation, which delineates defect shapes at the pixel level for detailed assessment (Rubio *et al.*, 2019).

Convolutional Neural Networks (CNNs) dominate due to their hierarchical feature extraction, delivering high accuracy in crack, spalling and corrosion detection (Li *et al.*, 2020[a]; Kruachottikul *et al.*, 2021; Zhang *et al.*, 2018). Variants such as VGG16 (Zhang *et al.*, 2024) FCNs (Li *et al.*, 2020[b]), Faster R-CNN (Kalfarisi *et al.*, 2020) and DenseNet (Lopez Droguett *et al.*, 2022) balance depth, speed and precision. Real-time systems like YOLOv8 achieve significant precision in crack detection, outperforming R-CNNs in both speed and accuracy (Xiong *et al.*, 2024), (Teng *et al.*, 2022).

Vision Transformers (ViTs) apply self-attention to image patches, capturing global patterns and excelling in complex defect recognition (Dosovitskiy *et al.*, 2020; Qi *et al.*, 2024). Swin Transformers add hierarchical, window-based attention for efficient high-resolution analysis (Liu *et al.*, 2021; Wan *et al.*, 2023). ConvNeXt, blending CNN and transformer insights, achieves remarkably high accuracy in bridge defect classification and robust pixel-level segmentation under varied conditions (Ma *et al.*, 2022; Pang *et al.*, 2024).

Training these models demands substantial computational resources, often employing multi-GPU cloud platforms (Ansari, 2020; Katiyar *et al.*, 2021; Talari *et al.*, 2023). Commercial services like Microsoft Custom Vision on Azure support rapid deployment via APIs and k-fold validation, simplifying ML model development at subscription cost (Ali and Ishak, 2020; Farley *et al.*, 2024).

Despite extensive research on concrete bridge defect detection, masonry bridges, a crucial component of the UK's railway infrastructure, remain underexplored. Saadatmorad *et al.* (2023) applied image processing techniques to masonry crack detection without rigorous ML validation. Loverdos and Sarhosis (2024) tested CNNs in laboratory settings, while Katsigiannis *et al.* (2023) focused solely on crack detection in building masonry. Consequently, there is a clear research gap in applying recent ML models to real-world 360° imagery of masonry bridges, encompassing a broader range of defects such as spalling, vegetation and loss of section.

This study assesses the applicability of contemporary ML models for automated detection of defects in 360° imagery of masonry bridges within operational environments.

2. Research method

The methodology employed in this research is illustrated in Figure 2, which comprises three primary steps. Initially, a literature review was conducted to identify the most appropriate ML approaches and models. The second step involved conducting experiments with the selected ML models, which included data preparation, model training and testing, and the analysis of the training and test results to evaluate the performance of the ML models in detecting defects in masonry bridges.

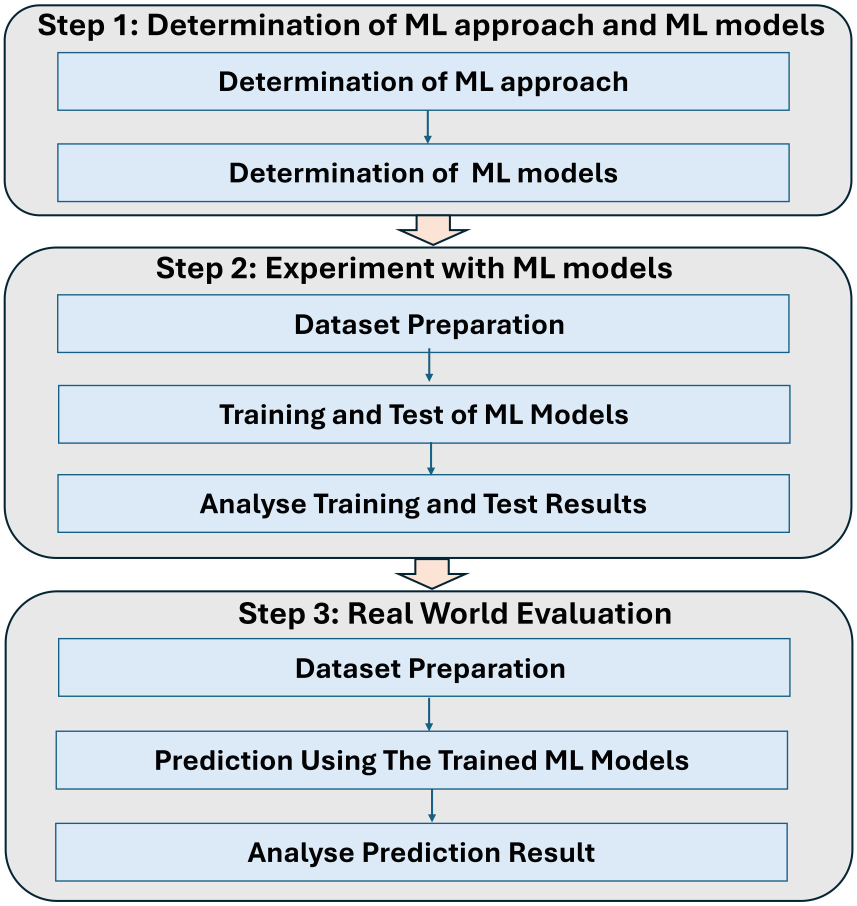

Step 1: Determination of ML approach and ML models
Determination of ML approach
Determination of ML models
Step 2: Experiment with ML models
Dataset Preparation
Training and Test of ML Models
Analyse Training and Test Results
Step 3: Real World Evaluation
Dataset Preparation
Prediction Using The Trained ML Models
Analyse Prediction Result

Fig. 2. Research methodology flowchart.

For ML model training and testing, sliced and labelled 360-degree images of masonry bridges were used. These images were provided by the asset owner. In the third step, the trained models were employed to predict defects in 360-degree images of several previously unseen bridges. The entire 360-degree image was sliced and labelled, and all image slices were input into the trained ML models to determine their predictions. The outcomes of these predictions were analysed to assess the feasibility and effectiveness of ML models for detecting masonry defects in real-world scenarios.

2.1 Determination of ML Approach

Key architectures such as Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), and hybrid architectures that combine both paradigms have been widely adopted for defect detection using structural imagery. Among these, CNNs remain the most extensively employed models for image-based defect detection in bridge structures. CNNs have been successfully applied to a variety of tasks, including classification (Deng *et al.*, 2021), object detection (Teng *et al.*, 2022), and segmentation (Rubio *et al.*, 2019). Over time, CNNs have undergone significant evolution, leading to the development of various architectures and variants designed to address challenges related to network depth, training efficiency, and prediction accuracy.

Prominent CNN architectures utilised in bridge defect detection include Fully Convolutional Networks (FCNs), Visual Geometry Group Networks (VGG-Net), Region-based Convolutional Neural Networks (R-CNN), Densely Connected Convolutional Networks (DenseNet), and You Only Look Once (YOLO). FCNs have demonstrated substantial advantages in defect detection tasks such as cracks (Li *et al.*, 2020^{b}), delamination, and exposed rebar (Rubio *et al.*, 2019), owing to their ability to generate pixel-level predictions. VGG-Net, particularly VGG16, is a well-established CNN model recognized for its architectural simplicity and robust performance in both image classification and object detection. These models can achieve up to 89% accuracy in detecting a wide range of bridge surface defects in concrete bridges (Zhang *et al.*, 2024), and when enhanced through transfer learning, their crack detection performance can exceed 97% with significantly reduced training time (Yang *et al.*, 2020). R-CNN and its derivatives have also been applied to defect detection which have achieved high accuracy in crack identification (Reghukumar and Anbarasi, 2021). The enhanced Faster R-CNN architecture improves detection speed, making it suitable for real-time applications, albeit with a minor trade-off in precision (Kalfarisi *et al.* , 2020).

DenseNet represents another influential CNN variant and for crack segmentation in concrete bridge, it has outperformed conventional semantic segmentation models (Lopez Droguett *et al.*, 2022). It has also exhibited superior classification performance compared to models like Xception, Inception, VGG, and ResNet (Akgül, 2023). Similarly, EfficientNet has introduced an innovative approach to model scaling by uniformly adjusting network width, depth, and resolution through fixed coefficients. This strategy enables high performance with improved computational efficiency (Tan and Le, 2019). In bridge defect detection, EfficientNet has demonstrated excellent results, surpassing DenseNet and Inception in identifying structural anomalies such as cracks and spalling (Zakaria, *et al.,* 2022).

The YOLO family of models represents a state-of-the-art, real-time object detection framework with multiple versions (YOLOv1–YOLOv8). YOLO models have shown higher detection accuracy than baseline VGG models (Chen, 2024). For crack detection in concrete bridges, YOLO outperforms R-CNN in both speed and accuracy (Deng *et al.*, 2021). In multi-defect detection scenarios involving cracks, swelling, and holes, Fast R-CNN performs slightly better than YOLOv3 (Jiang *et al.*, 2023). However, for detecting cracks and exposed rebar, YOLOv3 demonstrates superior speed and precision compared to Faster R-CNN (Teng *et al.,* 2022). The improved YOLOv4 further enhances both accuracy and processing speed (Yu, Shen and Shen, 2021), while YOLOv8 achieves over 98% precision in crack detection (Xiong *et al.*, 2024).

Recent advancements in computer vision have seen the emergence of Vision Transformers (ViTs), which adapt transformer architectures originally designed for natural language processing to visual tasks (Dosovitskiy *et al.*, 2020). Unlike CNNs, ViTs process images as sequences of patches that are linearly embedded, combined with positional encodings, and fed into transformer encoders employing self-attention mechanisms and multilayer perceptrons (Amirkhani *et al.*, 2024). This design enables ViTs to capture global contextual relationships within an image, enhancing their capacity to learn complex visual features. ViTs have demonstrated impressive performance in detecting defects such as cracks, patches, spalling, and corrosion on concrete and asphalt surfaces (Asadi Shamsabadi *et al.*, 2022).

The Swin Transformer, an advanced derivative of the ViT, represents a major step forward in computer vision, particularly for high-resolution and detail-sensitive tasks. Its hierarchical architecture allows for efficient processing of fine-grained textures, making it highly effective in detecting structural defects such as cracks and corrosion (Wan *et al.*, 2023). Enhanced versions of the Swin Transformer, incorporating additional layers and feature extraction modules, have been optimized for industrial surface defect detection (Zhou *et al.*, 2024; Gao *et al.*, 2022).

ConvNeXt, a modern CNN proposed by Liu *et al.* (2022), bridges the gap between classical convolutional models and transformer-based architectures. ConvNeXt has demonstrated superior performance in handling large input sizes, surpassing Swin Transformer in classifying concrete crack images (Zhang *et al.*, 2022). It achieved an accuracy of 99.22% in bridge defect detection, outperforming ResNet152 in recall, precision, and overall accuracy (Ma *et al.*, 2022). Moreover, ConvNeXt has proven effective for crack segmentation in bridges and pavements under challenging conditions such as uneven lighting and variable crack widths (Pang *et al.*, 2024; Zhao *et al.*, 2024)

Cloud-based solutions, such as Microsoft's Custom Vision within the Azure AI platform, offer accessible environments for developing and deploying custom CNN-based image classification and object detection models (Farley *et al.*, 2024). Using k-fold cross-validation to evaluate model performance (Maghraby, 2021), Custom Vision enables rapid, code-free model development (Malanca, 2020). Its applications extend to urban surveillance and image classification, including the detection of rescue vehicles in dense traffic (Ali and Ishak, 2020).

Research on concrete and steel bridges demonstrates the effectiveness of CNN based models in capturing both global and local image features (Zhang *et al.*, 2024; Lopez Droguett *et al.*, 2022). ConvNeXt's capability of capturing image features in diverse condition (Pang *et al.*, 2024; Zhao *et al.*, 2024) is crucial for masonry bridges with inherently irregular textures. FCNs and transformer models such as Swin Transformer are particularly suitable for detailed surface segmentation, enabling precise mapping of cracks and efflorescence (Li *et al.*, 2020; Wan *et al.*, 2023).Studies using VGG and EfficientNet further highlight the benefits of transfer learning, which can mitigate data scarcity challenges by fine-tuning pre-trained models (Yang *et al.*, 2020; Zakaria *et al.*, 2022). ViT and Swin Transformer architectures, with their superior ability to model complex spatial relationships (Zhou *et al.*, 2024; Gao *et al.*, 2022),hold potential for differentiating between natural masonry surface textures and genuine structural defects. Although limited research exists for steel structures, similar ML architectures have proven effective for detecting fatigue cracks (Dung *et al.*, 2019; Pang *et al.*, 2024).

Collectively, the literature review on defect detection in concrete and steel structures indicates a clear trajectory toward deep learning and transformer-based methodologies. These insights can inform the development of automated, high-precision defect detection systems for masonry bridges, where irregular textures and ageing materials pose unique challenges. Integrating CNNs, transformer-based global models, and transfer learning techniques can provide a robust foundation for domain-specific adaptation and real-world deployment.

This study selected the image classification approach over object detection for automated defect detection in railway bridge imagery. While object detection models can explicitly identify and localise defect regions within an image, they are computationally demanding and require more extensive annotation, as both class and bounding-box labels must be generated. In contrast, classification focuses on labelling smaller, pre-segmented image patches, improving training stability and computational efficiency. This approach, therefore, provides a practical and scalable means of assessing model feasibility within the available dataset. Although classification does not directly output spatial localisation of defects, this can be achieved by reassembling and visualising the classified patches within the 360° virtual environment, allowing detected defects to be viewed in their correct spatial context while maintaining the efficiency of the classification framework.

Based on the literature review four leading ML models were selected, representing a broad spectrum of classification approaches:

1. EfficientNet-B0: A classical convolutional neural network (CNN) architecture.

2. Swin Transformer - Tiny: A transformer-based model showcasing advanced feature extraction capabilities.
3. ConvNeXt-Tiny: A hybrid model combining the strengths of CNNs and transformers.
4. Custom Vision Compact: A commercially available solution on the Microsoft Azure platform.

These models were chosen to highlight the diversity of methodologies, encompassing traditional CNNs, cutting-edge transformers, hybrid architectures, and practical commercial solutions. Lightweight variants of each model were prioritised to facilitate rapid prototyping and computational efficiency, aligning with the overarching goal of validating the workflow rather than achieving peak performance. This strategy enabled a balance between theoretical robustness and operational feasibility for defect detection tasks.

2.2 Dataset Preparation

As discussed in Section 1.1, various types of defects can occur in masonry structures, some of which are more common than others. In this study, the six common defect types, namely Crack, Joint defect, Loss of Section, Spalling, Vegetation and one non defect class labelled as Other, were selected based on the reference documents provided by the asset owner. The classification of these defects was carried out according to the defect description provided in the reference documents supplied by the asset owner. Each image was annotated by trained engineers and researchers; the image slices were reviewed by the research team to assess their appropriateness for training ML models. Table 1 provides a summary of the dataset distribution across training (original and oversampled) and test sets.

Table 1: Summary of the dataset distribution.

Class	Training Set (Original)	Training Set (Oversampled)	Test Set
Crack	32	409	8
Joint Defect	244	409	61
Loss of Section	27	409	6
Other	409	409	102
Spalling	135	409	33
Vegetation	252	409	62

The dataset exhibited substantial class imbalance, with certain defect classes (e.g. Crack, Loss of Section) significantly underrepresented. Following an initial 80/20 train-test split that preserved natural class proportions in the test set, the training data were balanced using the Synthetic Minority Oversampling Technique (SMOTE) (Chawla *et al.*, 2002).

SMOTE was implemented via the imbalanced-learn library and applied separately to each class, increasing minority class samples through feature-space interpolation. All classes in the training set were adjusted to 409 samples, matching the original size of the majority class (Other). This ensured balanced class representation during training while maintaining an unbiased evaluation on the original test set.

3. ML Model Training (Experiment):

Three models, EfficientNet-B0, Swin Transformer-Tiny, and ConvNeXt-Tiny, were trained locally using the PyTorch framework with pretrained weights from torchvision.models. They were lightweight variants trained under the same local conditions, with each completing within approximately one hour. Experiments were conducted on a workstation with an AMD Ryzen 7 9800X3D CPU, 48 GB RAM, and an NVIDIA RTX 4080 (16 GB) GPU, using a Conda environment with Python 3.10.16 and PyTorch 2.5.1 (CUDA 12.4). All training configurations, data augmentations, and layer freezing strategies described below apply only to these locally trained models.

Data Augmentation: To enhance robustness and generalisation, the following augmentations were applied.

- Random cropping and resizing
- Horizontal and vertical flipping
- Brightness, contrast, and saturation adjustments

Training Configuration:

- Data split: 80% training / 20% validation
- Batch size: 32
- Random seed: 20 (for reproducibility)
- Optimiser: Adam
    - L2 regularisation: 0.001
    - L1 regularisation: 0.0
    - Learning rate: 0.0001
- Learning Rate Scheduler: ReduceLROnPlateau (factor = 0.5, patience = 2 epochs)
- Early Stopping: Patience = 5 epochs

Selective Layer Freezing:

- EfficientNet-B0: Stem and head frozen; MBConv blocks trainable

- Swin Transformer - Tiny: Stage 4 and final normalisation trainable; earlier stages frozen
- ConvNeXt-Tiny: Only Stage 4 trainable; preceding layers frozen.

The fourth model, Custom Vision Compact, was trained using Microsoft Azure's Custom Vision service under its default (auto-configured) settings. This cloud-based platform abstracts the training process, providing no access to internal parameters such as:

- Layer freezing or architecture modifications
- Custom data augmentation
- Learning rate or optimiser settings

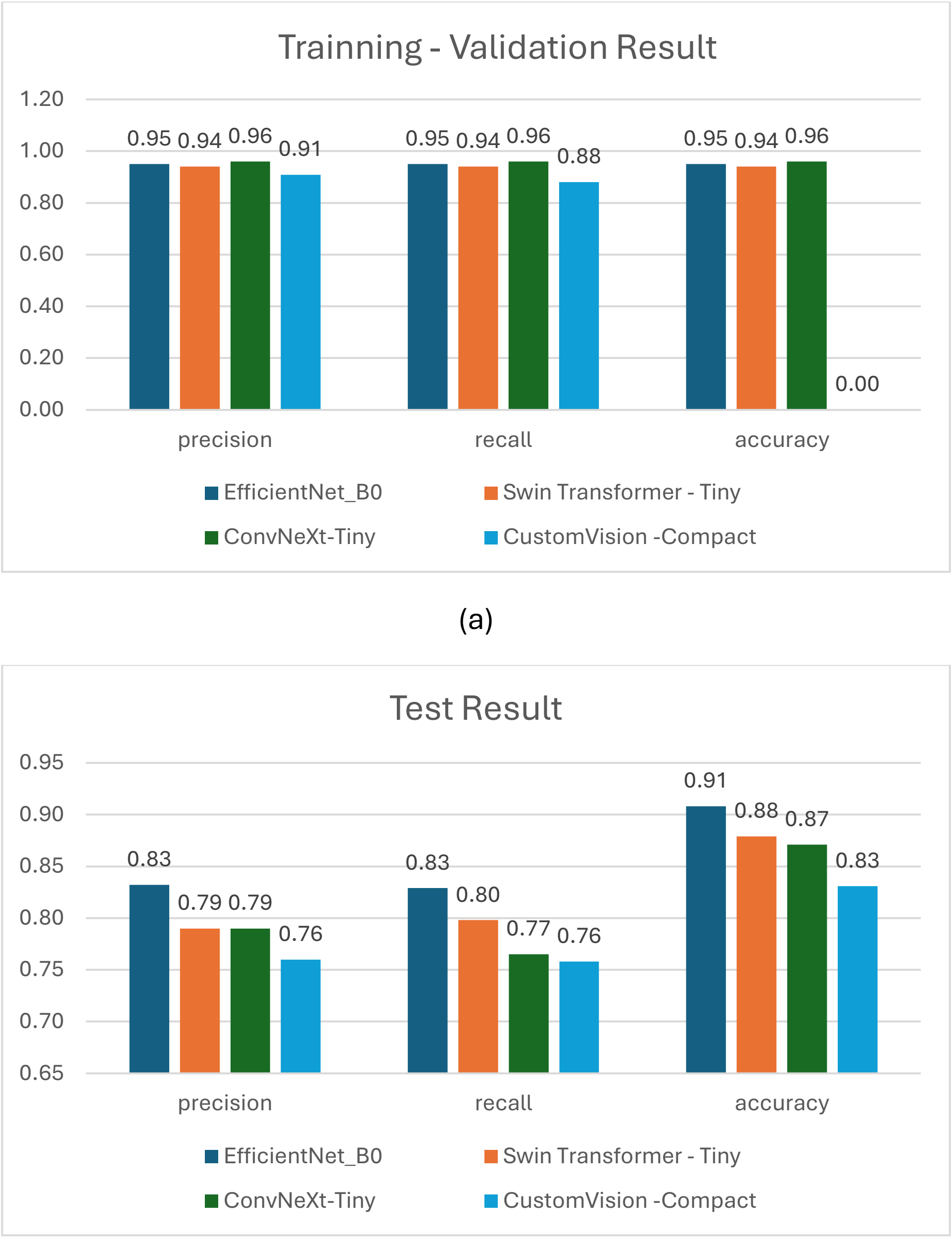
While the exact configuration is proprietary, Azure documentation indicates that the platform employs transfer learning with automated preprocessing and augmentation routines broadly similar to those applied in the local models (Microsoft, 2024). This model was included to assess the performance of a practical, deployable solution requiring minimal setup, albeit at the cost of transparency and fine-tuning control.

4. Experiment Results and Discussion:

4.1 Overall

The performance evaluation of the four ML models revealed consistent trends across precision, recall, and accuracy metrics during training and testing, confirming their applicability within the proposed workflow.

During training, all models achieved high precision, recall, and accuracy values (ranging from 0.91 to 0.96 - Note: CustomVision's training accuracy is recorded as 0.00 because the metric is not provided by the Azure service), indicating effective learning from the training dataset. However, a performance drop was observed during testing, reflecting the challenges posed by the imbalanced test dataset and real-world defect detection conditions. EfficientNet-B0 demonstrated the best generalisation, achieving the highest test accuracy of 0.91 and balanced precision and recall of 0.83 each. Its simpler, more balanced architecture makes it well-suited to this application, enabling robust performance despite the class imbalance in the test set.



(a)

(b)

Fig. 3. Performance of different ML models in the laboratory set up: (a) Training performance and (b) Test performance.

The other models—Swin Transformer - Tiny, ConvNeXt-Tiny, and Custom Vision Compact—exhibited a greater performance drop during testing. Testing accuracy for these models ranged from 0.83 to 0.88, with lower precision and recall compared to EfficientNet-B0. Figure 3 shows the training and test performance of different ML models.

The performance drop in Swin Transformer and ConvNeXt-Tiny can be attributed to their more complex architectures, which rely on advanced feature extraction mechanisms. While these models performed well during training, the layer-freezing strategy applied to optimise computational efficiency may have limited their ability to fully adapt to the dataset, contributing to reduced generalisation. Despite the use of early stopping, slight overfitting appears to have occurred, as indicated by the high training metrics. The Custom Vision Compact model, while practical and easy to deploy, lacks the flexibility for fine-tuning and advanced customisation, which may have further impacted its performance on the test set.

Overall, these results validate the effectiveness of integrating ML models into the digital bridge examination workflow. Among them, EfficientNet-B0 demonstrates the best overall balance between computational efficiency and classification performance under challenging conditions. However, since accuracy alone cannot fully reflect class-level behaviour in imbalanced data, a closer examination of F1 scores is required to understand each model's true discriminative capability.

4.2 Class-specific Analysis

While the models generally demonstrated effective overall performance on the test set according to weighted metrics, examining the class-specific F1 scores (Table 2 and Figure 4) provides crucial insights into how well each identified individual defect types, revealing performance variations influenced by several key factors:

Table 2: Detailed F1 scores

Model	Crack	Joint	Loss of Section	Other	Spalling	Vegetation	Macro Average	Weighted Average
EfficientNet-B0	0.67	0.93	0.67	0.92	0.82	1	0.83	0.91
Swin Transformer	0.59	0.9	0.67	0.88	0.74	1	0.79	0.88
ConvNeXt-Tiny	0.43	0.9	0.73	0.87	0.74	0.98	0.78	0.87
Custom Vision	0.62	0.84	0.57	0.84	0.67	0.96	0.76	0.83

i) Original data representation (Table 1): A primary driver of performance was the original sample count for each class before SMOTE balancing. Classes with larger initial representation ('Other' - 409, 'Joint Defect' - 244, 'Vegetation' - 252 samples) consistently achieved higher F1 scores (>0.84) on the test set.

ii) Inherent class characteristics: Visual distinctiveness significantly impacted results. The easily recognisable 'Vegetation' class achieved near-perfect F1 scores (0.96-1.00). Conversely, 'Spalling' (135 original samples) showed only moderate performance, likely hindered by potential visual ambiguity or overlap with other defects, making it intrinsically harder to classify. The subtle nature of features for other classes, like fine cracks, also likely contributed to classification difficulty beyond just sample size.

iii) Challenge of extreme rarity and SMOTE impact: The models consistently struggled with the classes having extremely low original counts: 'Crack' (32 samples) and 'Loss of Section' (27 samples) yielded the lowest F1 scores (dropping below 0.6 for some models). This poor performance persisted despite using SMOTE to balance the training data, suggesting the initial scarcity severely limited the ability to learn generalisable features for the imbalanced test set. This may indicate limitations in SMOTE's effectiveness in generating sufficiently informative samples in this complex image context when starting from such extreme underrepresentation.

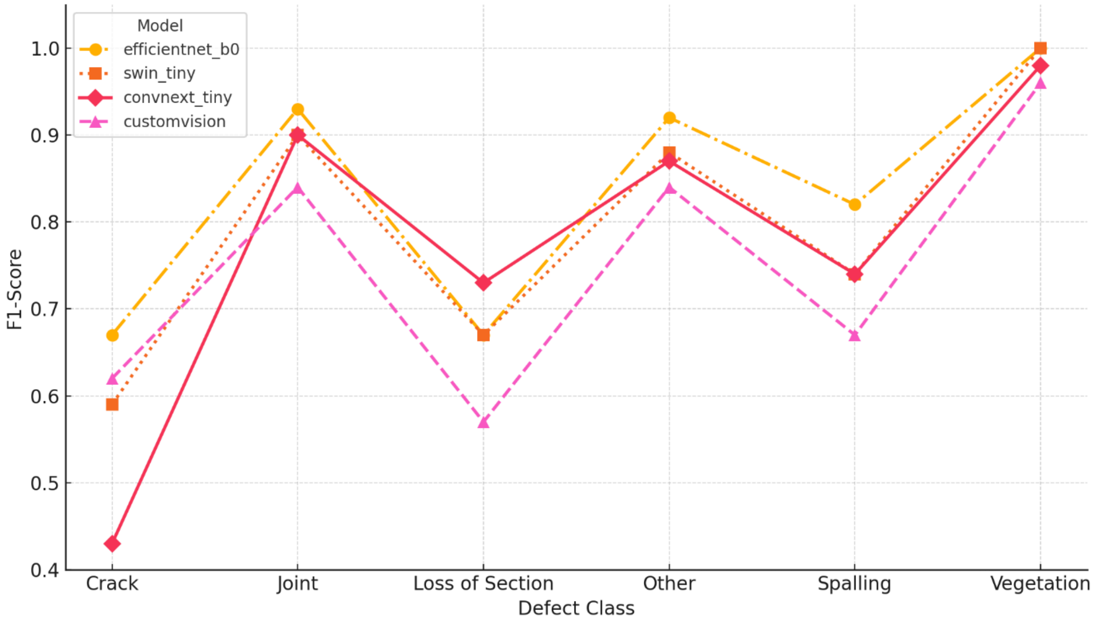


Fig. 4. F1 score by defect classes for different ML models in the laboratory set up.

5. Real-World Evaluation

5.1 Dataset Preparation

A new set of 360-degree images supplied by the owner of masonry bridges was used for the evaluation of the ML models' performance in a real-world environment. The images were captured using TLS and UAV cameras, and out of 49 images from 7 bridges, 9 such images were chosen to evaluate the prediction performance of the trained ML models in a real-world scenario. While selecting the images, the following two criteria were considered: (1) coverage of a bridge from different positions, i.e., all the images should not represent either underneath the arch or outside the arch, and (2) minimal obstructions from surrounding trees, plants, vegetation, scaffoldings for ongoing repairs, and any signposts. The reason for such image selection is to truly simulate real-world conditions, as in reality, some bridges may have a large number of defects while others may be defect-free. Although the number of images is limited and the class distribution is highly imbalanced, this reflects the actual conditions of masonry bridge inspections, where most areas show no visible deterioration and severe defects occur only locally. The purpose of this evaluation was therefore to assess model performance under realistic field distributions rather than to construct a statistically balanced test set. Table 3 provides detailed information about the case study bridges, including the number of images considered, positions, and the number of defects identified (true labels).

Table 3: Detailed information about the images considered for real-world evaluation.

Bridge	No. of images considered	Position	No. of defects (True label)
A	1	Underneath the arch	10 (9 Spalling and 1 Vegetation)
B	1	Underneath the arch	33 (2 Joint Defect, 1 Loss of Section, 28 Spalling and 2 Vegetation)
C	1	Underneath the arch	1 (joint)
D	1	Underneath the arch	79 (7 Crack, 51 Joint Defect and 21 Vegetation)
E	2	Outside	None
		Underneath the arch	2 Crack
F	2	Underneath the arch	6 (4 Joint Defect, 1 Spalling and 1 Vegetation)
			2 Spalling
G	1	Outside	5 (4 Spalling, 1 Vegetation)

The original 360-dgree images used for railway bridge defect detection are high-resolution, ranging from 10,000 x 5,000 to 20288 x 10144 pixels (Table 10). Processing these large images directly is computationally intensive and impractical. Therefore, they are typically sliced into smaller patches. The evaluation dataset was derived from 9 typical railway bridge 360-degree images, which were segmented into smaller patches of 224×224 pixels resulting in 30577 image patches. These patches were then annotated by trained engineers to ensure accurate labelling of defect types. The dataset is categorised into previously defined six defect classes, with the following distribution:

- Joint Defect: A minority class, with 58 images.
- Crack: Another minority class, containing 9 images.
- Loss of Section: The smallest and most underrepresented class, with only 1 image.
- Spalling: A small class, consisting of 44 images.

- Vegetation: Another small class, containing 26 images.
- Other: The dominant class, comprising 30,439 images, representing the vast majority of the dataset.

This dataset reflects the real-world distribution of defects in railway bridges, where the majority of areas may lack notable defects (labelled as "Other"). However, the severe class imbalance, with underrepresented defects like Crack, Loss of Section, and Spalling, poses significant challenges for ML models, which tend to prioritise the dominant class.

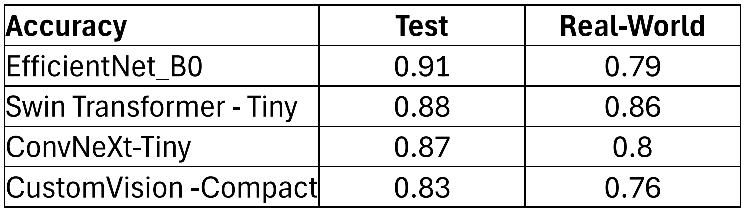
5.2 Result and Discussion

The performance of the four trained models was evaluated on this new dataset which reflects a realistic, highly imbalanced distribution of defect classes.

5.2.1 Overall Accuracy

A comparison of overall accuracy between the initial 'Test' set and the 'Real-World' dataset (Table 4) shows an expected performance drop for all models when faced with new data under realistic imbalance conditions.

Table 4: Comparison of ML Model Accuracy: Test vs Real-World.



Accuracy	Test	Real-World
EfficientNet_B0	0.91	0.79
Swin Transformer - Tiny	0.88	0.86
ConvNeXt-Tiny	0.87	0.8
CustomVision -Compact	0.83	0.76

The magnitude of this drop varied: Swin Transformer exhibited the most robustness in this metric, decreasing only slightly from 0.88 to 0.86 accuracy, while EfficientNet-B0 showed the largest decline, from 0.91 to 0.79. Consequently, Swin Transformer achieved the highest overall accuracy on the real-world data, followed by ConvNeXt-Tiny (0.80), EfficientNet-B0 (0.79), and CustomVision-Compact (0.76). However, given the severe class imbalance in the real-world dataset, where the 'Other' (no defect) class vastly outnumbers all defect classes, overall accuracy can be misleading. High accuracy might primarily reflect success in identifying the dominant 'Other' class, rather than effective defect detection. Therefore, a deeper analysis using class-specific metrics is essential.

5.2.2 Class-specific Performance

Examining the defect-specific F1 scores (Table 5) confirms the limitations suggested by the overall accuracy analysis. Compared to the initial test set results, there is a dramatic collapse in F1 scores for all actual defect classes across all four individual models when evaluated on the real-world dataset.

Table 5: Comparison of defect-wise ML Model F1 score - Test vs Real-World

F1 score	Model	Crack	Joint Defect	Loss of Section	Other	Spalling	Vegetation
Test	EfficientNet-B0	0.67	0.93	0.67	0.92	0.82	1
Real-World	EfficientNet-B0	0.03	0.07	0	0.88	0.02	0.01
Test	swin_tiny	0.59	0.9	0.67	0.88	0.74	1
Real-World	swin_tiny	0.04	0.07	0.01	0.92	0.04	0.02
Test	ConvNeXt-Tiny	0.43	0.9	0.73	0.87	0.74	0.98
Real-World	ConvNeXt-Tiny	0.02	0.04	0.01	0.89	0.03	0.02
Test	Custom Vision	0.62	0.84	0.57	0.84	0.67	0.96
Real-World	Custom Vision	0.01	0.02	0.01	0.86	0.02	0.02

Most F1 scores for defects in this scenario are near zero (often below 0.1), indicating very poor precision and recall combined. In stark contrast, the F1 score for the majority 'Other' class remains high (0.86-0.92). The Recall scores (Table 6) further illuminate the issue, revealing high false negative rates for most defect types. Models frequently fail to identify actual defects present in the images, particularly for 'Crack' and 'Spalling'. The 'Loss of Section' class presents a unique case: several models achieved 1.0 recall; however, there is only one loss of section presented in the dataset.

Table 6: Comparison of defect-wise ML Model recall - Test vs Real-World

Recall	Model	Crack	Joint Defect	Loss of Section	Other	Spalling	Vegetation
Test	EfficientNet-B0	0.63	0.9	0.67	0.9	0.88	1
Real-World	EfficientNet-B0	0.22	0.55	0	0.79	0.52	0.81
Test	Swin_tiny	0.63	0.93	0.67	0.86	0.7	1
Real-World	Swin_tiny	0.33	0.66	1	0.86	0.48	0.85
Test	ConvNeXt-Tiny	0.38	0.92	0.67	0.84	0.79	1
Real-World	ConvNeXt-Tiny	0.33	0.69	1	0.8	0.5	0.81
Test	Custom Vision	0.5	0.92	0.67	0.76	0.73	0.97
Real-World	Custom Vision	0.11	0.53	1	0.76	0.34	0.69

The near-zero F1 score for this class suggests this correct identification was likely accompanied by numerous false positives, leading to extremely low precision. EfficientNet-B0 failed to detect even the single instance (0 recall). While recall for the 'Other' class was reasonably high, it was generally lower than its F1 score, indicating some non-defective patches were incorrectly classified as defects (contributing to false positives for defect classes).

5.2.3 Ensemble Model Performance

To assess if combining prediction results could improve performance, an ensemble model was formulated based on a majority vote across the four individual models. (For tie-breaking where models split evenly, e.g., two predict Class A and two predict Class B, the prediction from the EfficientNet-B0 model was used). The ensemble results, shown in Table 7 for the real-world dataset, demonstrate some marginal gains.

Table 7: Recall and F1 score for the ensembled model.

Metric	Crack	Joint Defect	Loss of Section	Other	Spalling	Vegetation
F1 score	0.24	0.19	0.09	0.97	0.1	0.04
Recall	0.22	0.64	1	0.95	0.52	0.85

While F1 and recall scores for 'Crack' and 'Joint Defect' saw slight improvements over the average individual model, they remained extremely low (e.g., F1 scores of 0.24 and 0.19 respectively). The ensemble achieved its best performance, unsurprisingly, on the dominant 'Other' class (F1=0.97, Recall=0.95). The fact that even an ensemble approach struggles significantly confirms that the primary challenge lies not just in model selection but in the fundamental difficulty of detecting rare and sometimes visually subtle defects within a highly imbalanced dataset reflecting real-world conditions. Without strategies specifically targeting this imbalance and the potential visual ambiguity between defect classes and background textures, the models evaluated here consistently overlook critical defects. From a practical perspective, this indicates that ensemble modelling, while theoretically beneficial, offers limited value for operational deployment. The marginal improvements achieved do not justify the additional computational cost and implementation complexity, suggesting that streamlined individual models may provide a more efficient and equally reliable option for integration into inspection workflows.

5.2.4 Misclassifications Analysis

Given the significant drop in performance observed for defect classes in the real-world evaluation, particularly the low F1 and recall scores for rare defects, a detailed misclassification analysis is necessary. The purpose of this analysis is twofold: first, to pinpoint specific confusion patterns between defect classes (i.e., which defects are commonly mistaken for others), and second, to quantify the impact of the severe class imbalance, particularly the dominance of the 'Other' class, on the models' predictions.

To achieve a broad understanding of common error tendencies that persist across the different model architectures evaluated, the analysis presented here aggregates the predictions from all four individual models (EfficientNet-B0, Swin Transformer, ConvNeXt-Tiny, Custom Vision) and the resulting ensemble model. This combined approach helps identify fundamental challenges in distinguishing certain defects within this dataset. The following tables break down these aggregated misclassifications:

- Table 8 (percentage misclassification table): Shows the proportion of each true class that was predicted as different categories. This helps identify the most frequent types of confusion.
- Table 9 (real count misclassification table): Provides the actual number of instances where a true class was predicted as another class across all five model runs combined. This illustrates the scale of misclassifications, especially involving the large number of 'Other' class instances.
-

Table 8: Proportion(%) of true class predicted as different class for each defect (five models together).

Defect type	**Crack**	**Joint Defect**	**Loss of Section**	**Other**	**Spalling**	**Vegetation**
Crack	24.44	37.78	2.22	33.33	2.22	0.00
Joint Defect	1.38	61.38	5.17	27.24	1.72	3.10
Loss of Section	0.00	0.00	80.00	0.00	20.00	0.00
Other	0.64	4.80	0.57	83.18	4.43	6.38
Spalling	0.91	7.27	9.09	32.27	47.27	3.18
Vegetation	0.00	0.00	0.77	19.23	0.00	80.00

Table 9: Number of true defects predicted as different defects (five models together).

Defect type	**Real Count**	**Total Count**	**Crack**	**Joint Defect**	**Loss of Section**	**Other**	**Spalling**	**Vegetation**
Crack	9	45	11	17	1	15	1	0
Joint Defect	58	290	4	178	15	79	5	9
Loss of Section	1	5	0	0	4	0	1	0
Other	30439	152195	980	7300	865	126597	6736	9717
Spalling	44	220	2	16	20	71	104	7
Vegetation	26	130	0	0	1	25	0	104

- Crack: Crack is heavily confused with Joint Defect (38%) and "Other" (33%), meaning the model struggles to differentiate fine crack patterns from joint defects. The low true count (9 cases) makes these errors significant. Although 24% of Crack cases are correctly classified, the majority are lost to Joint Defect or "Other". Minor misclassifications into Loss of Section and Spalling (~2%) indicate occasional overlap with other structural damage.
- Joint Defect: The model correctly identifies Joint Defect 61% of the time, but 27% of cases are misclassified as "Other", meaning many real Joint Defect cases go undetected. Additionally, 5% of cases are misclassified as Loss of Section, which suggests visual similarities between these two damage types. The model also occasionally confuses Joint Defect with Vegetation and Spalling, though at lower rates.
- Loss of Section: This class is extremely rare (1 real instance), making meaningful conclusions difficult. However, the model predicts Loss of Section far more frequently than it actually occurs (5 total predictions), leading to false positives from other defects. No cases were correctly classified, meaning the model lacks sufficient examples to generalize effectively.
- Other: The most stable class (83% correct), but its dominance means even small misclassification percentages translate to large numbers. The biggest misclassification issue is with Vegetation (6%), suggesting some background textures or blended patterns in real-world images contribute to confusion. Joint Defect is also occasionally over-predicted within "Other" (~5%), though the impact is smaller.
- Spalling: The model struggles with Spalling, as more than half of the cases are misclassified (only 47% correct). 32% of Spalling cases are lost to "Other," while 9% are mistaken for Loss of Section. These misclassifications suggest that Spalling shares structural damage patterns with multiple defect types, making classification more challenging.
- Vegetation: Vegetation is mostly well-recognised (80%), but 19% of cases are misclassified as "Other." This suggests that in real-world settings, some vegetation cases are harder to distinguish from background elements, leading to classification errors. Other misclassifications are minimal, indicating that Vegetation is relatively distinct compared to other classes.

In summary, this detailed misclassification analysis highlights several fundamental challenges in automatically identifying masonry defects in this real-world dataset. Key issues include frequent confusion between visually similar defect types (such as 'Crack' and 'Joint Defect'), significant misclassification of actual defects as 'Other' (non-defect background), contributing to missed detections, and the inherent difficulty models face in reliably identifying extremely rare classes like 'Crack' and 'Loss of Section' even when aggregated across multiple architectures. These challenges underscore the limitations revealed by the performance metrics in the previous section.

5.3 Bridge-wise Performance Analysis

Out of 138 defects, 85 (61%) defects are correctly predicted by the ensemble model, which comprises 2 Cracks, 37 Joint Defects, 1 Loss of Section, 23 instances of Spalling, and 22 instances of Vegetation. Out of 30,439 images that are not defects (i.e., belong to the "Other" class), 1,637 (5%) image slices are incorrectly identified as defects by the ensemble model. Among these, 6 are Cracks, 286 are Joint Defects, 388 are instances of Spalling, 14 are Loss of Sections, and 943 are instances of Vegetation. To analyse the performance of the ML predictions across different bridges, a bridge-wise analysis was conducted based on the segmented 360-degree image patches. Table 10 evaluates the performance of the ensemble model in predicting defects for each bridge, comparing the true labels with the correct predictions.

The analysis of the ensemble model's performance across different bridges and defect types supports the key findings of the misclassification analysis. The model struggled with predicting Cracks in both Bridge B and Bridge E, where the true cracks were found. For Joint Defects, the prediction is better than for Cracks, with around 62% correctly predicted in Bridge D, where most of the Joint Defects occurred.

Table 10: Bridge-wise performance of ensembled model for different defect classes.

Bridge	Image file name and size in pixels	Number of defects in an image	Crack	Joints Defect	Loss of Section	Spalling	Vegetation
A	Image 01 (18440 X 9220)	True label- 10	-	-	-	9	1
		Correct prediction - 1	-	-	-	0	1
B	Image 01 (20288 X 10144)	True label- 33	-	2	1	28	2
		Correct prediction- 24	-	1	1	20	2
C	Image 01 (10324 X 4268)	True label - 1	-	1	-	-	-
		Correct prediction - 1	-	1	-	-	-
D	Image 01 (20288 X 10144)	True label-79	7	51	-	-	21
		Correct prediction - 51	2	32	-	-	17
E	Image 01	None					

Bridge	Image file name and size in pixels	Number of defects in an image	Crack	Joints Defect	Loss of Section	Spalling	Vegetation
	(20288 X 10144)						
	Image 02 (20288 X 10144)	True label- 2	2	-	-	-	-
		Correct prediction - 0	0	-	-	-	-
F	Image 01 (20288 X 10144)	True label- 6	-	4	-	1	1
		Correct prediction - 4	-	3	-	0	1
	Image 02 (20288 X 10144)	True label- 2	-	-	-	2	-
		Correct prediction - 0	-	-	-	0	-
G	Image 01 (10000 X 5000)	True label - 5	-	-	-	4	1
		Correct prediction - 4	-	-	-	3	1

In Bridges B, C, and F, the number of true defects is very low, ranging between 1 and 4, making it difficult to reach a reasonable conclusion. Loss of section in Bridge B is correctly predicted, but with only one instance, making it difficult to reach a reliable conclusion. Predicting Spalling defects is reasonably good in Bridges B and G, but the rest are not detected, specifically in Bridge A, where none of the 9 Spalling defects were detected. The model demonstrates higher accuracy in predicting Vegetation across all bridges.

6. Conclusion:

This study presents a new examination of masonry bridges by deploying ML techniques for the automatic detection of defects in 360-degree field imagery of operational structures. Unlike prior work confined to laboratory settings and prototype systems, this research evaluates representative ML architectures under real-world conditions, marking a significant advance in field-scale masonry bridge inspection.

Across EfficientNet-B0, ConvNeXt-Tiny, Swin Transformer-Tiny and a Microsoft Azure Custom Vision model, performance was broadly consistent during training and controlled testing, demonstrating that these typical architectures share similar baseline capabilities. However, when transferred to highly imbalanced, varied field images, precision and recall for critical defects, such as Cracks, Loss of Section and Spalling, dropped dramatically, with F1 scores often below 0.1. Including an explicit “Other” category was necessary to reflect the predominance of intact masonry surfaces, yet it exacerbated misclassification by absorbing as much as 32% of true Spalling cases and 27% of Joint Defects. A detailed error analysis further revealed systematic confusion between defect classes, with approximately 38% of Cracks mislabelled as Joint Defects and rare categories exhibiting high false-positive rates due to limited training examples. Vegetation detection remained relatively robust but still suffered occasional misclassifications against complex background textures.

The ensemble approach provided only marginal improvements over individual models, confirming that model selection alone is insufficient to overcome the challenges posed by real-world data imbalance and inter-class visual similarity. Bridge-wise analysis further indicated that defect detection performance varied across individual structures, suggesting that contextual factors and bridge-specific characteristics may influence model effectiveness.

Building on the detailed quantitative analyses, three additional insights emerge:

- In controlled experiments with a SMOTE-balanced test set, EfficientNet-B0 led all architectures, achieving 0.91 accuracy and balanced precision and recall of 0.83.

- Even after oversampling, extremely rare classes exhibited persistently low F1 scores, indicating that synthetic augmentation alone cannot fully compensate for scarce examples.

- Among the models evaluated, the Swin Transformer-Tiny exhibited the smallest performance drop from experiment to field conditions (declining from 0.88 to 0.86 accuracy), suggesting that certain architectures may better tolerate real-world variability.

In summary, this study demonstrates the feasibility of applying machine learning for automated defect detection in masonry bridges using 360° imagery, providing an important step towards digitised and data-driven inspection workflows. The evaluated models achieved strong performance in controlled conditions and revealed the key factors that influence their generalisation to field data.

Although the patch-based visualisation of classification results already enables defects to be viewed in their correct spatial context within the 360° virtual environment, further post-processing could help merge adjacent or related patches to produce more continuous and interpretable representations. From a practical perspective, these findings indicate that compact ML models can support preliminary screening and consistency in bridge assessments, reducing manual workload and inspection time. However, reliable deployment in operational settings will require the integration of such models into existing inspection procedures, ensuring that automated outputs complement, rather than replace, expert judgement. Future research should focus on developing larger, balanced datasets, enhancing classification through spatial coherence and domain knowledge, and validating approaches with expert-reviewed ground truth to advance effective asset management.

Acknowledgement:

The authors would like to acknowledge the Network Rail R&D programme for providing image data. The contents of this publication are the sole responsibility of its authors and cannot be taken to reflect the views of Network Rail.

Reference :

Abdallah, A.M., Atadero, R.A. and Ozbek, M.E. (2022) ‘A State-of-the-Art Review of Bridge Inspection Planning: Current Situation and Future Needs’, *Journal of Bridge Engineering*, 27(2). doi:10.1061/(ASCE)BE.1943-5592.0001812.

Akgül, İ. (2023) ‘Mobile-DenseNet: Detection of building concrete surface cracks using a new fusion technique based on deep learning’, *Heliyon*, 9(10). doi:10.1016/j.heliyon.2023.e21097.

Ali, O. and Ishak, M.K. (2020) ‘Bringing intelligence to IoT Edge: Machine Learning based Smart City Image Classification using Microsoft Azure IoT and Custom Vision’, *Journal of Physics: Conference Series*, 1529(4), p. 042076. doi:10.1088/1742-6596/1529/4/042076.

Amirkhani, D. *et al.* (2024) ‘Visual Concrete Bridge Defect Classification and Detection Using Deep Learning: A Systematic Review’, *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–23. doi:10.1109/TITS.2024.3365296.

Ansari, S. (2020) ‘Computer Vision Modeling on the Cloud BT - Building Computer Vision Applications Using Artificial Neural Networks: With Step-by-Step Examples in OpenCV and TensorFlow with Python’, in Ansari, S. (ed.). Berkeley, CA: Apress, pp. 389–442. doi:10.1007/978-1-4842-5887-3_10.

Asadi Shamsabadi, E. *et al.* (2022) ‘Vision transformer-based autonomous crack detection on asphalt and concrete surfaces’, *Automation in Construction*, 140(April), p. 104316. doi:10.1016/j.autcon.2022.104316.

Chawla, N. V. *et al.* (2002) ‘SMOTE: Synthetic Minority Over-sampling Technique’, *Journal of Artificial Intelligence Research*, 16(1), pp. 321–357. doi:10.1613/jair.953.

Chen, S. *et al.* (2019) ‘UAV Bridge Inspection through Evaluated 3D Reconstructions’, *Journal of Bridge Engineering*, 24(4), pp. 1–15. doi:10.1061/(ASCE)BE.1943-5592.0001343.

Chen, Y. (2024) ‘The Investigation of Performance Comparison for VGG, YOLO, and DINO in Image Classification’, *Highlights in Science, Engineering and Technology*, 85, pp. 984–990. doi:10.54097/9bgem219.

Chow, J.K. *et al.* (2021) ‘Automated defect inspection of concrete structures’, *Automation in Construction*, 132(August), p. 103959. doi:10.1016/j.autcon.2021.103959.

Deng, J., Lu, Y. and Lee, V.C.-S. (2021) ‘Imaging-based crack detection on concrete surfaces using You Only Look Once network', *Structural Health Monitoring*, 20(2), pp. 484–499. doi:10.1177/1475921720938486.

Dosovitskiy, A. *et al.* (2020) 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. doi:https://doi.org/10.48550/arXiv.2010.11929.

Dung, C.V. *et al.* (2019) 'A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks', *Automation in Construction*, 102(March), pp. 217–229. doi:10.1016/j.autcon.2019.02.013.

Farley, P., Mehrotra, N. and Urban, E. (2024) *What is Custom Vision?* Available at: https://learn.microsoft.com/en-us/azure/ai-services/custom-vision-service/overview (Accessed: 26 August 2024).

Gao, L. *et al.* (2022) 'Cas-VSwin transformer: A variant swin transformer for surface-defect detection', *Computers in Industry*, 140, p. 103689. doi:10.1016/j.compind.2022.103689.

Humpe, A. (2020) 'Bridge Inspection with an Off-the-Shelf 360° Camera Drone', *Drones*, 4(4), p. 67. doi:10.3390/drones4040067.

Jiang, S. *et al.* (2023) 'Automatic Detection of Surface Defects on Underwater Pile-Pier of Bridges Based on Image Fusion and Deep Learning', *Structural Control and Health Monitoring*. Edited by Y.-Q. Ni, 2023, pp. 1–17. doi:10.1155/2023/8429099.

Kalfarisi, R., Wu, Z.Y. and Soh, K. (2020) 'Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated Visualization', *Journal of Computing in Civil Engineering*, 34(3), pp. 1–20. doi:10.1061/(ASCE)CP.1943-5487.0000890.

Katiyar, A., Behal, S. and Singh, J. (2021) 'Automated defect detection in physical components using machine learning', *Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development, INDIACom 2021*, pp. 527–532. doi:10.1109/INDIACom51348.2021.00094.

Katsigiannis, S. *et al.* (2023) 'Deep learning for crack detection on masonry façades using limited data and transfer learning', *Journal of Building Engineering*, 76(March), p. 107105. doi:10.1016/j.jobe.2023.107105.

Kruachottikul, P. *et al.* (2021) 'Deep learning-based visual defect-inspection system for reinforced concrete bridge substructure: a case of Thailand's department of highways', *Journal of Civil Structural Health Monitoring*, 11(4), pp. 949–965. doi:10.1007/s13349-021-00490-z.

Li, G. *et al.* (2020^{b}) 'Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive Bayes data fusion based on a visual detection system', *Measurement Science and Technology*, 31(7), p. 075403. doi:10.1088/1361-6501/ab79c8.

Li, H. *et al.* (2020^{a}) 'Bridge Crack Detection Based on SSENets', *Applied Sciences*, 10(12),

p. 4230. doi:10.3390/app10124230.

Liu, Z. *et al.* (2021) ‘Swin Transformer: Hierarchical Vision Transformer using Shifted Windows’, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, pp. 9992–10002. doi:10.1109/ICCV48922.2021.00986.

Liu, Z. *et al.* (2022) *A ConvNet for the 2020s*. doi:https://doi.org/10.48550/arXiv.2201.03545. Available online at: https://arxiv.org/abs/2201.03545. Accessed on 14/11/2025.

Lopez Droguett, E. *et al.* (2022) ‘Semantic segmentation model for crack images from concrete bridges for mobile devices’, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 236(4), pp. 570–583. doi:10.1177/1748006X20965111.

Loverdos, D. and Sarhosis, V. (2024) ‘Pixel-level block classification and crack detection from 3D reconstruction models of masonry structures using convolutional neural networks’, *Engineering Structures*, 310(April), p. 118113. doi:10.1016/j.engstruct.2024.118113.

Ma, H. *et al.* (2022) ‘Parallel Systems for the Bridge Inspection’, *IEEE Journal of Radio Frequency Identification*, 6, pp. 783–786. doi:10.1109/JRFID.2022.3212598.

Maghraby, A. El (2021) ‘Improving Custom Vision cognitive services model’, *Journal of the ACS Advances in Computer Science*, 12(1), pp. 36–63. doi:10.21608/asc.2021.240134.

Majtan, E., Cunningham, L.S. and Rogers, B.D. (2023) ‘Numerical study on the structural response of a masonry arch bridge subject to flood flow and debris impact’, *Structures*, 48(August 2022), pp. 782–797. doi:10.1016/j.istruc.2022.12.100.

Malanca, A. (2020) *An experiment with Azure Custom Vision*. Available at: https://telefonicatech.uk/blog/an-experiment-with-azure-custom-vision/ (Accessed: 26 August 2024).

Masciotta, M.G. *et al.* (2023) ‘Integration of Laser Scanning Technologies and 360° Photography for the Digital Documentation and Management of Cultural Heritage Buildings’, *International Journal of Architectural Heritage*, 17(1), pp. 56–75. doi:10.1080/15583058.2022.2069062.

Meegoda, J.N., Kewalramani, J.A. and Saravanan, A. (2019) ‘Adapting 360-Degree Cameras for Culvert Inspection: Case Study’, *Journal of Pipeline Systems Engineering and Practice*, 10(1). doi:10.1061/(ASCE)PS.1949-1204.0000352.

Microsoft (2024) *Use cases for Custom Vision*. Available at: https://learn.microsoft.com/en-us/legal/cognitive-services/custom-vision/custom-vision-cvs-transparency-note (Accessed: 6 June 2025).

Mirzazade, A. *et al.* (2021) ‘Workflow for Off-Site Bridge Inspection Using Automatic Damage Detection-Case Study of the Pahtajokk Bridge’, *Remote Sensing*, 13(14), p. 2665.

doi:10.3390/rs13142665.

National Highways (2021) *Asset Data Management Manual*. Available at: https://nationalhighways.co.uk/media/fmrfnz1k/admmv13_part_2_requirements_and_additional_information_final.pdf (Accessed: 17 June 2023).

Network Rail (2018) *How structural inspections improve asset management*. Available at: https://www.networkrail.co.uk/stories/how-structural-inspections-improve-asset-management/#:~:text=Most are based on 'visual,photographed and its condition recorded. (Accessed: 15 July 2024).

Noy, E.A. and Douglas, J. (2005) *Building Surveys and Reports*. 3rd edn. Edited by M. Malden. Oxford: Blackwell Publication.

Omer, M. *et al.* (2021) 'Inspection of Concrete Bridge Structures: Case Study Comparing Conventional Techniques with a Virtual Reality Approach', *Journal of Bridge Engineering*, 26(10), pp. 1–13. doi:10.1061/(asce)be.1943-5592.0001759.

Pang, R. *et al.* (2024) *A Novel SegNet Model for Crack Image Semantic Segmentation in Bridge Inspection*, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Nature Singapore. doi:10.1007/978-981-97-2259-4_26.

Phares, B.M. *et al.* (2004) 'Routine Highway Bridge Inspection Condition Documentation Accuracy and Reliability', *Journal of Bridge Engineering*, 9(4), pp. 403–413. doi:10.1061/(asce)1084-0702(2004)9:4(403).

Qi, H. *et al.* (2024) 'A Vision-Transformer-Based Convex Variational Network for Bridge Pavement Defect Segmentation', *IEEE Transactions on Intelligent Transportation Systems*, PP, pp. 1–13. doi:10.1109/TITS.2024.3385788.

Reghukumar, A. and Anbarasi, L.J. (2021) 'Crack Detection in Concrete Structures Using Image Processing and Deep Learning', in *Sustainability*, pp. 211–219. doi:10.1007/978-981-15-9019-1_19.

Rubio, J.J. *et al.* (2019) 'Multi-class structural damage segmentation using fully convolutional networks', *Computers in Industry*, 112, p. 103121. doi:10.1016/j.compind.2019.08.002.

Saadatmorad, M. *et al.* (2023) 'Crack detection in historical masonry structures using efficient image processing: Application on a masonry bridge in Iran', *2023 IEEE International Workshop on Metrology for Living Environment, MetroLivEnv 2023 - Proceedings*, pp. 230–235. doi:10.1109/MetroLivEnv56897.2023.10164038.

Sen, A., Wu, S. and Talebi, S. (2025) 'Masonry Bridge Inspection Using Point Cloud Data and 360-Degree Images: A Study of Railway Bridges', *Journal of Performance of Constructed Facilities*, 39(5). doi:10.1061/JPCFEV.CFENG-4871.

Talari, V.S. *et al.* (2023) 'ENHANCING STRUCTURAL HEALTH MONITORING AND

MANAGEMENT THROUGH EDGE, FOG AND CLOUD COMPUTING ARCHITECTURES', in *Proceedings of the 14th International Workshop on Structural Health Monitoring*. Destech Publications, Inc., pp. 1537–1544. doi:10.12783/shm2023/36902.

Talebi, S. *et al*. (2022) 'The development of a digitally enhanced visual inspection framework for masonry bridges in the UK', *Construction Innovation*, 22(3), pp. 624–646. doi:10.1108/CI-10-2021-0201.

Tan, M. and Le, Q. V. (2019) 'EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks', in Chaudhuri, K. and Salakhutdinov, R. (eds) *Proceedings of the 36th International Conference on Machine Learning*. PMLR, pp. 6105--6114. doi:https://doi.org/10.48550/arXiv.1905.11946.

Teng, S., Liu, Z. and Li, X. (2022) 'Improved YOLOv3-Based Bridge Surface Defect Detection by Combining High- and Low-Resolution Feature Images', *Buildings*, 12(8), p. 1225. doi:10.3390/buildings12081225.

Wan, H. *et al*. (2023) 'A novel transformer model for surface damage detection and cognition of concrete bridges', *Expert Systems with Applications*, 213(PB), p. 119019. doi:10.1016/j.eswa.2022.119019.

Washer, G. *et al*. (2016) 'New Framework for Risk-Based Inspection of Highway Bridges', *Journal of Bridge Engineering*, 21(4), pp. 1–8. doi:10.1061/(ASCE)BE.1943-5592.0000818.

Wells, J. and Lovelace, B. (2017) *Unmanned Aircraft System Bridge Inspection Demonstration Project Phase II Final Report (No. MN/RC 2017-18)*. Available at: https://rosap.ntl.bts.gov/view/dot/32636.

Wells, J. and Lovelace, B. (2021) *Unmanned Aircraft Systems (UAS) – Metro District Bridge Inspection Implementation*. Minnesota. Dept. of Transportation. Office of Policy Analysis, Research & Innovation.

Xiong, C., Zayed, T. and Abdelkader, E.M. (2024) 'A novel YOLOv8-GAM-Wise-IoU model for automated detection of bridge surface cracks', *Construction and Building Materials*, 414(September 2023), p. 135025. doi:10.1016/j.conbuildmat.2024.135025.

Yang, Q. *et al*. (2020) 'Deep convolution neural network-based transfer learning method for civil infrastructure crack detection', *Automation in Construction*, 116(March), p. 103199. doi:10.1016/j.autcon.2020.103199.

Yu, Z., Shen, Y. and Shen, C. (2021) 'A real-time detection approach for bridge cracks based on YOLOv4-FPM', *Automation in Construction*, 122(January 2020), p. 103514. doi:10.1016/j.autcon.2020.103514.

Zakaria, M., Karaaslan, E. and Catbas, F.N. (2022) 'Advanced bridge visual inspection using real-time machine learning in edge devices', *Advances in Bridge Engineering*, 3(1), p. 27. doi:10.1186/s43251-022-00073-y.

Zekkos, A.A. *et al*. (2020) *Asset Management for Retaining Walls*, *Michigan Department of*

Transportation. Available at: https://www.michigan.gov/mdot/0,4616,7-151-9622_11045_24249-533554--,00.html.

Zhang, C., Chang, C.C. and Jamshidi, M. (2018) *Bridge Damage Detection using a Single-Stage Detector and Field Inspection Images*. Available at: http://arxiv.org/abs/1812.10590 (Accessed: 15 May 2024).

Zhang, H. *et al.* (2022) 'Crack Detection based on Convnext and Normalization', *Journal of Physics: Conference Series*, 2289(1). doi:10.1088/1742-6596/2289/1/012022.

Zhang, H. *et al.* (2024) 'Deep learning-based automatic classification of three-level surface information in bridge inspection', *Computer-Aided Civil and Infrastructure Engineering*, 39(10), pp. 1431–1451. doi:10.1111/mice.13117.

Zhao, M. *et al.* (2024) 'An Automated Instance Segmentation Method for Crack Detection Integrated with CrackMover Data Augmentation', *Sensors*, 24(2). doi:10.3390/s24020446.

Zhou, X. *et al.* (2024) 'A shunted-swin transformer for surface defect detection in roller bearings', *Measurement: Journal of the International Measurement Confederation*, 238(52275091), p. 115283. doi:10.1016/j.measurement.2024.115283.