Journal of Civil Structural Health Monitoring (2024) 14:103–120
https://doi.org/10.1007/s13349-022-00643-8

ORIGINAL PAPER

Artificial intelligence-based visual inspection system for structural health monitoring of cultural heritage

Mayank Mishra[1] · Tanmoy Barman[1] · G. V. Ramana[2]

Received: 28 May 2022 / Accepted: 15 October 2022 / Published online: 31 October 2022

Abstract

The United Nations aims to preserve, evaluate, and conserve cultural heritage (CH) structures as part of sustainable development. The design life expectancy of many CH structures is slowly approaching its end. It is thus imperative to conduct frequent visual inspections of CH structures following conservation guidelines to ensure their structural integrity. This study implements a custom defect detection, and localization supervised deep learning model based on the you only look once (YOLO) v5 real-time object detection algorithm by implementing a case study of the Dadi-Poti tombs in Hauz Khas Village, New Delhi. The custom YOLOv5 model is trained to automatically detect four defects, namely, discoloration, exposed bricks, cracks, and spalling, and tested on a dataset comprising 10291 images. The validity and performance of the custom YOLOv5 model are compared with a ResNet 101 architecture-based faster region-based convolutional neural network (R-CNN), and conventional manual visual inspection methods are used to convey the significance of the developed artificial intelligence-based model. The maximum average precision (mAP) of the custom YOLOv5 model and faster R-CNN is 93.7% and 85.1%, respectively.

Keywords Deep learning · Cultural heritage · Structural health monitoring · Convolution neural network · Classification · You only look once (YOLO) · Computer vision

1 Introduction

Authorities tasked with preserving historical sites use the visual inspection method to evaluate the state of preservation of cultural heritage (CH) structures and provide appropriate inputs for planning repair works. Damage assessments are mostly conducted when the severity of damage is evident from visual inspection and remedial measures become mandatory. Rapid urbanization, air pollution leading to discoloration of CH assets, vibrations, fires, humidity, high solar radiation, prevailing winds, floods, storms, and vandalism are all-natural and societal elements. These variables wreak havoc on CH by altering their inherent character and producing discoloration, abrasion, efflorescence, spalling, fissures, stains, and fungal development.

Traditional visual inspection employs highly skilled and experienced inspectors who document findings on-site using only their observations. Inspection staff visually evaluate the building, record each problem on paper or electronic devices, and photograph the flaws to validate the recording based on the codal standards and templates. Due to the structure and architecture of heritage sites, it is cumbersome to regularly perform an inspection of the entire CH [1]. Moreover, manual assessment is time-consuming and expensive due to the high labor input [2].

Innovative tools for inspecting CH structures are being utilized in combination with the traditional method of visual inspection such as laser scanning [3–6] and internet-of-things based sensors [7]. While equipment-based techniques benefit data operation and maintenance, almost all types of equipment have substantial installation and operating costs. This technology is fragile and difficult to install and deploy when investigating large and complex heritage structures,

✉ Mayank Mishra
mayank@iitbbs.ac.in; mayank_mishra@outlook.in

Tanmoy Barman
tb14@iitbbs.ac.in

G. V. Ramana
ramana@civil.iitd.ac.in

[1] School of Infrastructure, Indian Institute of Technology Bhubaneswar, Argul, Khordha, Odisha 752050, India

[2] Department of Civil Engineering, Indian Institute of Technology, Delhi 110016, India

and it requires prior permission from authorities to inspect CH structures. Following data gathering, the data must be further processed and analyzed.

Data processing and analysis require highly qualified and experienced personnel and specialized equipment; both are costly and cannot provide timely data for heritage structure inspection. Thus, primary assessments of CH structures take longer and are not performed frequently [8]. Many researchers have attempted to address the limitations of manual inspections by utilizing computer vision-based technologies such as digital image correlation (DIC) [9–11] and digital image processing (DIP) [12–15]. These methods rely on manual feature extraction and are affected by the lighting conditions. New inspection systems are required to aid conservation authority workers in expediting the inspection process and restoring historic structures. Developing automated digital systems, tools, and technologies is the focus of the present phase of the industry 4.0 revolution [16, 17]. One of the newest digital technologies in historic conservation is artificial intelligence (AI)-based automation [18–20]. Using deep learning (DL)-based convolutional neural networks (CNNs) to detect defects in photographs, AI can help overcome manual, and machine-based inspection limitations [21]. With advancements in technologies and computer science, new and improved techniques must be adopted for detecting damage in heritage structures.

The research work addresses the problem of CH assets inspection, which is a traditionally human-intensive activity. The AI-based model can provide ease for human effort in continuous monitoring of these CH and help identify defects in complex architectures by direct expert-based observations to aid the SHM process. This study proposes a new inspection method based on modern technologies and digital data to identify damage in CH structures. In particular, the damage is identified from a set of images focused on the various parts of the tombs considered. Since it is a real-time detection model, it can be used in conjunction with video captured from UAVs and mobile cameras. The pathologies of damages are identified by bounding boxes displaying probability values for the different classifications. The results obtained using the developed method are used for identifying suitable approaches for the preventive conservation and SHM of CH structures.

2 Previous related works

Machine learning (ML) techniques have been utilized to evaluate the structural health condition of CH buildings and identification of structural components in CH buildings [22–27]. Yao et al. [28] deployed the YOLOv4 network model for real-time, detection of concrete surface cracks. Their basic model can process 16 frames per second (FPS), which does not suffice for real-time but their YOLOv4 tiny network and improved YOLO achieved processing rates of 56 and 44 FPS, meeting the requirements of real-time damage detection. The current research addresses such visual-inspection systems, which can even aid manual inspection at locations not accessible by SHM inspection professionals. Furthermore, the research used all the image data gathered from the CH building, rather than using partial damage data and partial synthetic datasets generated from physics-based models in various researches [29, 30].

Some research works are carried out for CH inspection at various levels of SHM, ranging from defect detection in CH to detecting missing CH assets. Mansuri and Patel [31, 32] developed an automatic defect detection technique using a faster region-based CNN (R-CNN) model and Inception v2 DL architecture. A dataset of 880 images was considered, containing three types of surface defects: exposed brickworks, spalling damage, and cracks. This dataset was annotated using LabelImg software [33] for use as a ground truth images dataset for learning defects in images. The system can detect defects in CH structures with the highest detection accuracy (maximum average precision, mAP) of 0.915. Chaiyasarn et al. [34] deployed CNN to detect cracks in historical structures on dataset collected by camera and unmanned aerial vehicle (UAV). Dais et al. [35] used DL techniques for detecting cracks in masonry surfaces with complex backgrounds. Wang et al. [36] used CNN-based classification techniques with the sliding window algorithm to identify and locate different damage classes in historic masonry structures. In a structure in China that houses a historical museum, Wang et al. [37] used faster R-CNN to identify and detect several damage types from 100 roof images. Guo et al. [38] applied a rule-based Mask R-CNN model for assessing plastered and painted facade defects such as cracks, peeling, spalling, biological growth, delimitation and blistering in CH. Monna et al. [39] applied a combination of Faster R-CNN with artificial data augmentation to detect vernacular buildings from satellite imaginary with a correlation R^2 of 0.88 in one best-settings of the classifier. Wang et al. [1] developed a smartphone-based damage detection system that uses ML techniques to detect spalling and efflorescence in brick masonry walls in real-time. Mondal et al. [40] used R-CNN to classify four forms of earthquake-related damage, including exposed rebars, cracks, buckling, and spalling, in data collected from earthquake-damaged buildings. They used finite element techniques to develop a numerical model for localizing post-earthquake damage in masonry structures. Sharma et al. [41] used CNN to detect the quantity of dust deposited on heritage structures and to determine the level of damage caused. Bouchama [42] used deep CNN to detect damage caused by a variety of pathologies that can affect the surfaces of historic structures. Zou et al. [43] used CNN as an intelligent inspection system so that they could detect missing components in CH buildings. Conservators who conduct routine inspections of historic structures are particularly interested in these missing components. Masrour et al. [44] pre-trained DL-CNN models with transfer learning for detecting seven damage pathologies in old buildings in Morocco.

Automated approaches based on CNNs with different DL frameworks have been extensively used for detecting cracks in concrete [45, 46] and metal structures [47, 48]. These approaches can detect intrinsic details that cannot be observed visually. Drone-based inspection techniques can be used to detect structural problems in inaccessible areas, such as rooftops. Zhou et al. [49] deployed an R-CNN algorithm to detect cracks in crane steel structures in which data collection was conducted using a UAV with a detection accuracy of 95.4%. Most contemporary research is based on damage detection in concrete datasets, specifically for two types of images, cracked and uncracked. Kung et al. [50] also employed unmanned aerial vehicles (UAV) and DL to detect building deterioration due to surface defects with high precision and recall values. Chen et al. [51] divided the training dataset of images of concrete spaces (227×227×3) into cracked and uncracked images (20000 images of each type). The network was batch normalized after each pooling layer, and a large convolution kernel and pooling size were used [51]. The program automatically stopped training after 110,000 iterations, and model success rate was 99.71%. Cha et al. [52] used a DL technique for object detection and image classification. First, a rectangular box enclosure was used to detect irregularly shaped cracks. Simultaneously, window sliding strategies and region splitting were used. Semantic segmentation (e.g., FCN) was used to classify each image at the pixel level based on cracked and uncracked regions [53]. They proposed a U-net network structure based on FCN, which uses fewer training images and provides better results for some FCN metrics; they compared it with Cha's CNN method. Deng et al. [54] used the object detection network YOLO v2 to detect cracks in concrete with complex backgrounds (handwritten scripts), achieving a maximum mAP of 77%. Feng et al. [55] proposed an STDD network, a DL-based real-time exposed rebar detection method for spillway tunnels, that is efficient and lightweight. The STDD network outperformed SDDNet (a crack detection network in particular), with 1.7 million parameters and a 14.08 ms average inference time.

The literature review revealed two critical roadblocks in detecting defects in CH structures. First, most applications focus on concrete crack identification instead of CH defects. Secondly, application typology is the first thing to be considered in CH, as damage typologies in CH vary a lot. A typology is the application of defect detection, namely, the type of structure and components thereof. Some studies, for example, have only focused on detecting defects in brick walls, while some only damages due to dust deposition. These models failed to discover defects in the other components of the CH structure, such as a column or dome. It is hard to generalize one kind of defect among all structural elements because defects are unique to structural elements. Previous studies have been limited to a particular type of heritage structure and cannot be applied to other structures. Moreover, carrying out SHM must not be limited to a few expert individuals but should be easily carried out by someone with basic knowledge about SHM.

3 Data collection and preprocessing

The CH image dataset is collected from the Dadi-Poti tombs in Hauz Khas Village, New Delhi (Fig. 1). The larger tomb (Dadi) belongs to the Lodi era (1451-1526 AD), and measures 15.86 m × 15.86 m. The tomb has Quranic inscriptions engraved on the walls and ceiling in the form of medallions. The smaller tomb (Poti) is located 20 feet away. It belongs to the Tughlaq era (1321-1414 AD), measuring 11.8 m × 11.8 m.

Like all DL methods, proper supervision of labeled data with corresponding damage classes, image quality, sufficient images of each damage class and their quality, and tuning the DL model to perform for the custom dataset is challenging. In many cases, researchers have leaned on the generation of synthetic datasets and mixed several images with objects from the real world to populate their datasets. Still, we have only used the real-image datasets for our study. The collected image dataset comprises 3500 photographs of the Dadi and Poti tombs, which reveal various flaws, including spalling, efflorescence, exposed bricks, cracks, discoloration, algae growth, and missing parts. In this study, four of these classes of defects are identified and analyzed as there is fewer data available corresponding to the other defects. The images are annotated by drawing bounding boxes using roboflow software (Fig. 2). The annotated images are then augmented to a total of 10291 images by flipping, cropping, gray scaling, and rotation, as illustrated in Fig. 3, and the final dataset contains a total of 14757 labels, of which 975, 5896, 7752, and 134 are crack, discoloration, exposed brick, and spalling labels, respectively, as shown in Fig. 4. These 3500 original images were then shrunk to 416×416 pixels to reduce the training time.

For training the defect detection system, first, the defects available in the image should be manually identified using expert feedback from CH inspectors. Ground truth images are used to learn about image defects through manual identification. All features and information in a picture must be identified, and rectangular boxes must be drawn around each CH defect typology. The bounding box around each fault is carefully identified in all photos. For

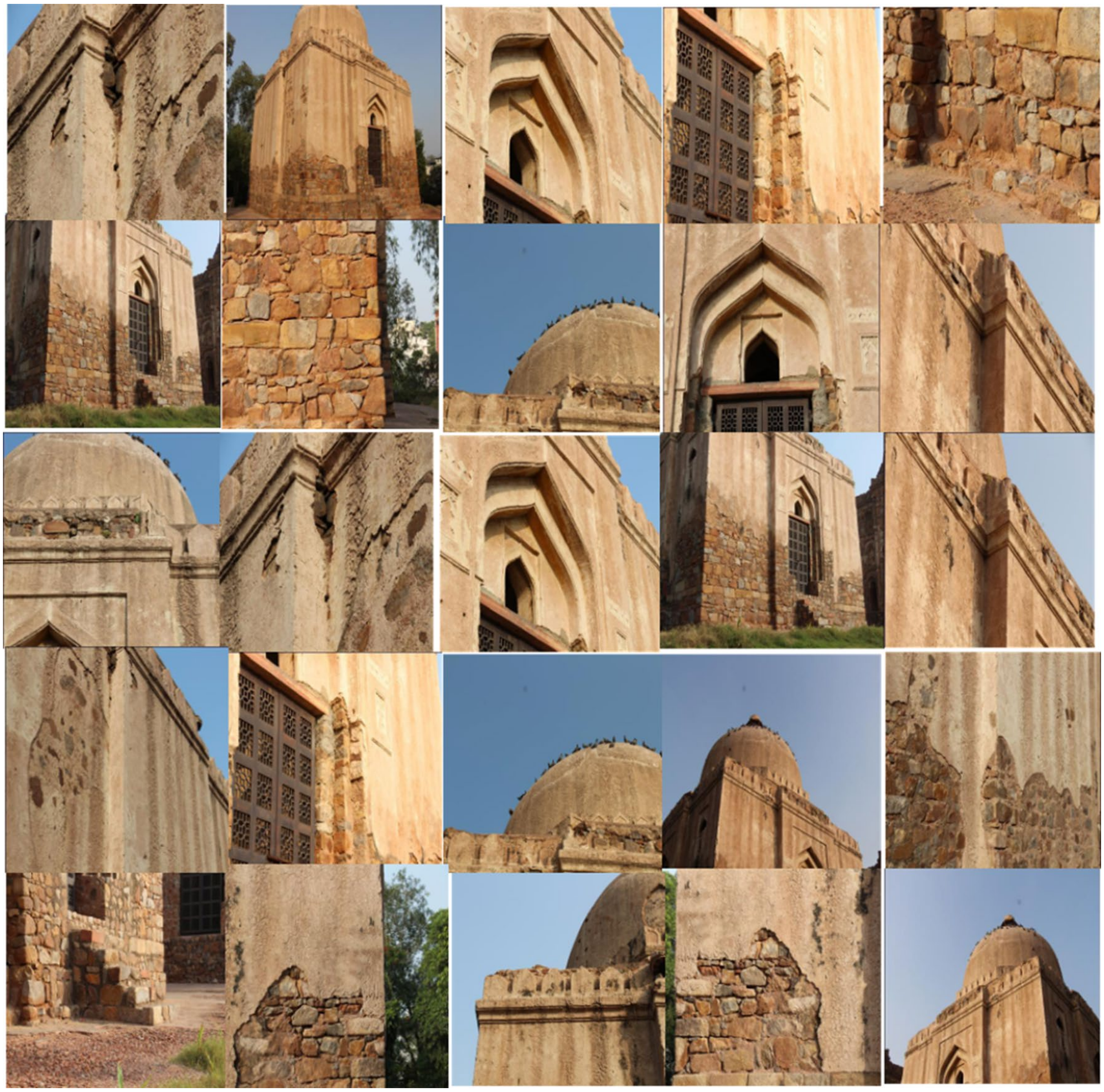


Fig. 1 Data collected from the Dadi-Poti tombs, Hauz Khas, New Delhi [96]

example, many images may have multiple bounding boxes for capturing various types of defects.

However, some limitations and problems still exist in model preparation, such as (1) the lack of availability of a sufficient amount of data to train the DL model for each class of defects. (2) time spent on cropping the images, splitting them and then augmenting them, (3) choosing the best algorithm for the particular case study in terms of detecting defects accurately as well as providing real-time or quasi-real-time defect detection, (4) image dataset containing instances of shadows and windows, and some of the data did not contain any prominent defects, so the model testing gave inaccurate predictions. Additionally, images containing no defects were marked as null and not discarded (which improves accuracy), and further augmentation, such as gray scaling.

4 Methodology

A DL advanced object detection model named YOLO [56, 57], is used to train a DL model on collected datasets. The method followed in this research is described below, and a schematic diagram of the steps is shown in Fig. 5. YOLO and its several versions have been successfully used in several object-detection applications in civil engineering such as structural crack detection [58–60], crack detection on concrete structures [28, 61–63] and pavement maintenance

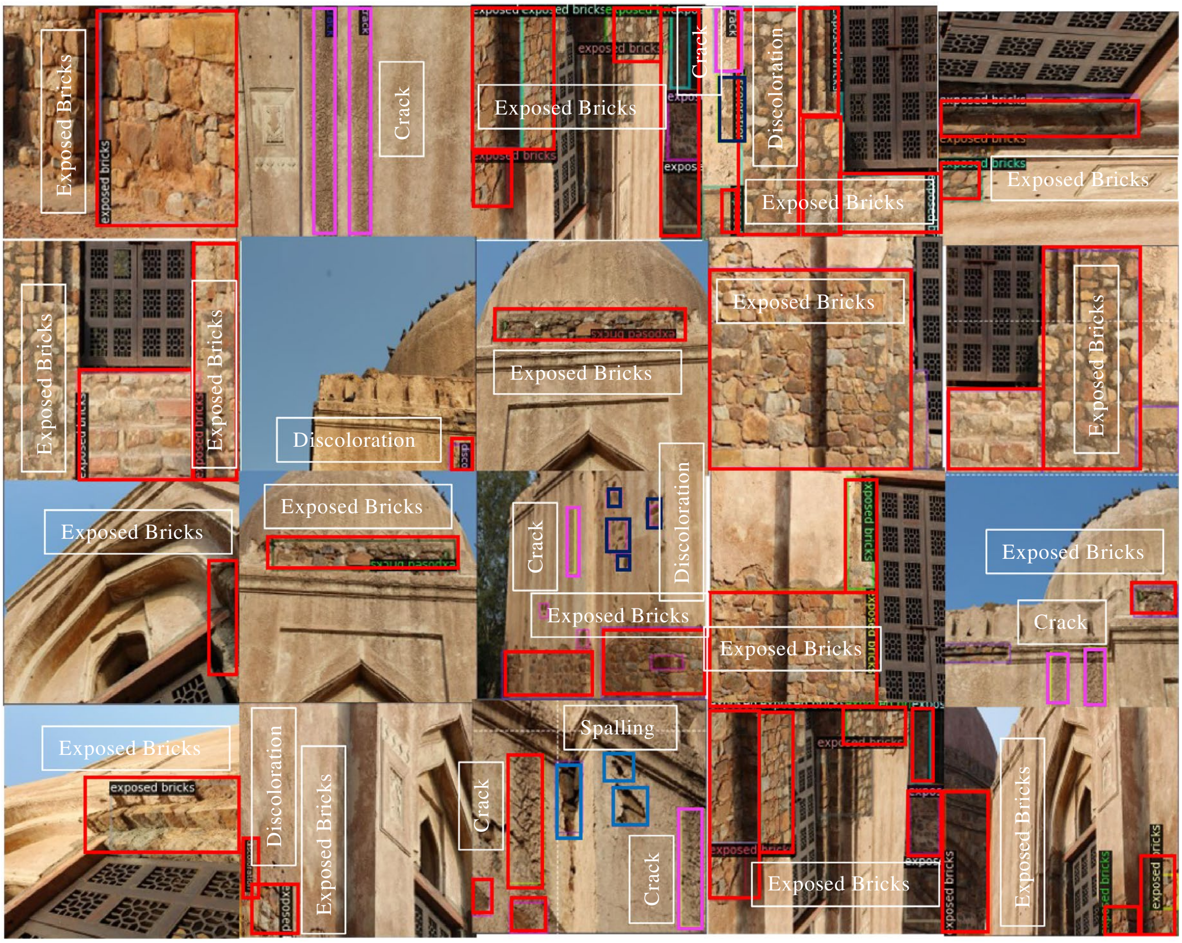


Fig. 2 Examples of data annotation. The data were annotated to identify four types of defects; spalling, crack, exposed brick, and discoloration

[64–67], spilled loads in traffic scene [68], identifying the parts of a building [69] and potholes-detection in pavement [70, 71], traffic load distribution in bridge infrastructure [72], safety-helmet detection in construction sites [73, 74], counting steel-pipes in construction sites [75], measure diameter of reinforcement bars [76], loosening of bolts in wind turbines [77], recognising structural components from 2D drawings [78] etc. YOLO is able to respond in real-time, even on devices with limited processing power, since it uses only one forward propagation through the neural network to make a forecast. An application of YOLO in CH automatic detection of spalling zones in limestone walls was performed by Idjaton et al. [79] on 1000 high-resolution images of CH in France.

4.1 Convolutional neural networks

A CNN comprises an input layer, a convolutional layer, a pooled layer, an input layer (from flattening), a fully connected layer, and an output layer. A convolution comprises three elements over an operation: an input image, a feature detector (kernel), and a feature map. The feature detector filters the information in the input image. It filters the integral parts, excluding the rest, to get the corresponding feature map. The decrease in input image size depends on the size of the kernel and the number of strides taken in traversing the pixels. CNN develops multiple feature detectors, developing several feature maps called a convolutional layer. The next component in the first step of CNN is the rectified linear unit (ReLU). ReLU increases the non-linearity of images as they are naturally non-linear. Therefore, the rectifier further breaks the linearity to compensate for the linearity imposed on an image while it is undergoing convolution.

4.2 Pooling layer

There are various types of pooling, including mean, max, and random. In this study, max pooling is used. It enables the CNN to detect certain features, specifically, the maximum value in the pooling area. We extract the maximum value to account for distortions, i.e., to provide CNN with spatial variance capability. Pooling also decreases image sizes

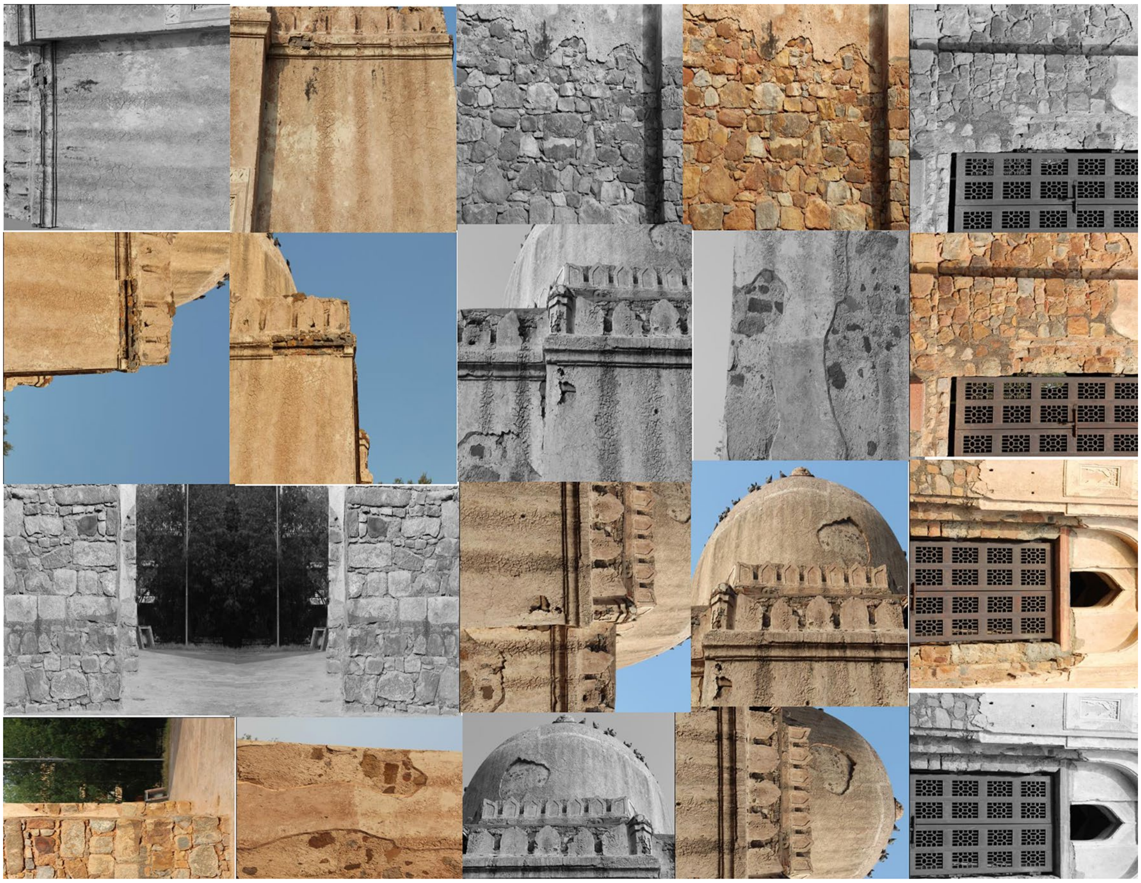


Fig. 3 Examples of data augmentation in CH images. Augmentation includes gray scaling, flipping, rotation and cropping of images

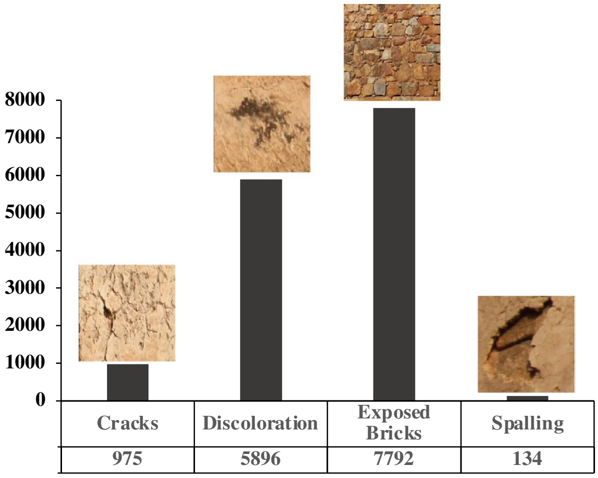


Fig. 4 Bar graph representing the total instances of defects in each class in the dataset with 10291 images

and the number of parameters, thus preventing overfitting. The pooled feature map is then inserted into an artificial neural network as an input layer after flattening it. Meanwhile, the network's building blocks, such as the weights and the feature maps, are trained and continuously updated to achieve optimal performance, enabling it to classify images and objects with maximum accuracy. Each pooling layer prevents overfitting, hence expediting convergence and enhancing training stability.

4.3 Adam algorithm

The Adam algorithm is an optimizer for model training because it can perform stochastic gradient descent to set a learning rate for each weight parameter [80]. A lower learning rate is provided for weights updated more frequently, and a higher learning rate is provided for weights updated the least often while changing parameters. The algorithm can also provide a lower learning rate for weights that are updated less frequently.

4.4 Implementation

The current research is focused on constructing an AI-based state-of-the-art defect detection algorithm that leverages DL. However, the development of such a model necessitates the use of advanced DL techniques for object detection. The

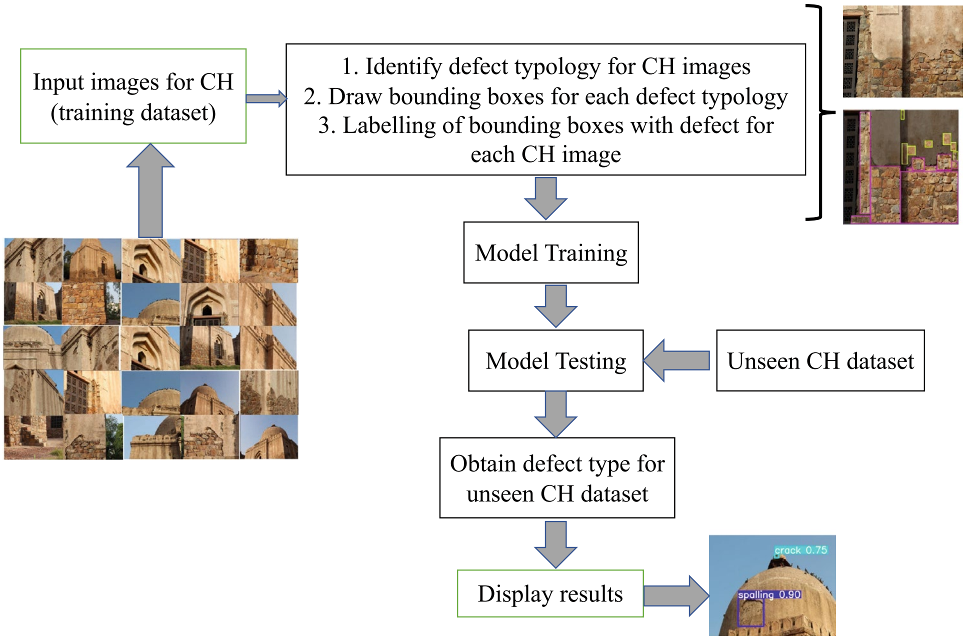

Input images for CH (training dataset)
1. Identify defect typology for CH images
2. Draw bounding boxes for each defect typology
3. Labelling of bounding boxes with defect for each CH image
Model Training
Model Testing
Unseen CH dataset
Obtain defect type for unseen CH dataset
Display results
crack 0.75
spalling 0.90

Fig. 5 Overall methodology of implementing defect detection models based on DL

YOLOv5 method [81] is used in this project to develop a new object detection model for detecting four distinct defects.

Validation of the model can be inferred from the accuracy with which bounding boxes are detected. Moreover, any model requires comparison with its peers for validation and to test its robustness. The model selected for comparison is the state-of-the-art, commonly used deep learning-based method named, faster R-CNN [82, 83] based on the ResNet 101 architecture.

Originally, YOLOv5 was a single-stage object detector with three main parts, i.e., backbone, neck, and head. The model backbone comprises cross-stage partial (CSP) networks that are used to extract essential features from the input image. The neck is mainly used to generate feature pyramids that help models generalize on object scaling. It helps identify the same object in different sizes and scales. For the most part, the head is used for final detection. Finally, it produces final output vectors that include class probabilities, objectness scores, and bounding boxes for each feature.

Thus, an object detection model based on the state-of-the-art object detection method, the YOLOv5 algorithm, is developed whereby the CNNs of the YOLO network is designed to give the best performance on the image dataset. The C3 convolution layer is replaced by bottleneck CSP networks, further improving the model's performance.

First, the images are tagged using roboflow online software. A total of 3550 photos are labeled to train the model to identify four different types of flaws, such as spalling, discoloration, exposed bricks, and cracks. This dataset is then divided into three subsets: a training set, a test set, and a validation set, each with a 80:10:10 split [84, 85]. To expedite the learning process, photos are resized to 416 × 416 pixels during the preprocessing stage. A total of 10291 photos are added to the dataset post-labeling. Flipping, cropping, gray scaling, and a 90° rotation are used to improve the dataset. After training on the COCO dataset [86], the batch size is set to 16, the number of epochs is 50, and pre-trained weights are used.

The training dataset is fed into the custom YOLO model with pre-trained weights and hyperparameters tuned in to give the best performance. Here, the custom YOLO model means that it is optimized for our particular application and the custom data poti dataset. In the custom YOLOv5s network used, this C3 convolution network is replaced with a bottleneck CSP layer that is used to make residual blocks thinner to increase depth and have fewer parameters, to increase computation speed as well as accuracy. The YOLOv5 has four types of model architecture that are used for different sizes of datasets, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The YOLOv5s is suitable for a smaller dataset and requires considerably less computation time than the YOLOv5l, which requires a large dataset and higher computation time but gives highly accurate results when the prior two conditions are satisfied. The performance of the custom YOLOv5s and YOLOv5l models is evaluated, and mAP is calculated to be 93.7% for the custom YOLOv5s model. In contrast, the YOLOv5l model does not give satisfactory results and requires three times the computation time. The YOLOv5s model is initially trained for 50 epochs, as higher epochs mean higher processing time, and the free version of Google Collaboratory (Google Colab) gives runtime disconnect errors. Each iteration's best and last weights are saved and used while running the code for the subsequent 50 epochs. Higher values of epochs do not entirely change the accuracy, and it saturates over 90 epochs up to 100 epochs.

The dataset is trained with custom YOLOv5 algorithms, namely, custom YOLOv5s and custom YOLOv5l. Compared to the YOLOv5s model, the YOLOv5l model has more parameters, requires more CUDA memory to train, and is slower. Thus, after training on the dataset a few times, the latter is discarded, and YOLOv5s, as shown in Fig 6, is adopted. The YOLOv5s algorithm is deployed because it is fast and provides real-time detection of defects. It contains a total of 191 layers, 7.46816 million parameters, and gradients. After confirming the correctness of the model, it is selected to be used for the principal dataset, as shown in the Fig. 6, following the same procedure as used for the custom dataset. The performance of the model is improved by providing more data, optimizing the hyperparameters, and using different weights.

4.5 Environment

Google Collab is an integrated development environment (IDE) that supports research and learning related to AI. Collab provides a code environment similar to Jupyter Notebook, and it offers a free graphics processing unit (GPU) and a tensor processing unit (TPU). Google Collab has popular pre-installed libraries such as PyTorch, TensorFlow [87], Keras [88], and OpenCV. ML or DL algorithms require systems with high speed and processing power (usually based on a GPU); standard computers are not equipped with a GPU, and buying one is expensive. Hence, Google Collab supplies GPUs (Tesla V100) and TPUs (TPUv2) over the cloud to assist AI researchers.

4.6 Performance metrics

Based on relevance, the performance of the custom YOLOv5 is evaluated using the parameters of precision and recall. Precision, also known as a positive predictive value, and recall, also known as sensitivity, are given using Eqs. 1 and 2, respectively. The robustness of the proposed YOLOv5 algorithm is measured using the value of AP, which is the area between the precision and recall curves. A single value is obtained for AP, which shows that the detector can classify correctly and identify all objects that fit these classifications. The mAP is calculated by taking each class's average of the AP values.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

where TP, FP, TN, and FN denote true positive, false positive, true negative, and false negative, respectively.

4.7 Loss function

The YOLOv5 network is trained using stochastic gradient descent with momentum (SGDM), a training optimization approach that helps accelerate gradient vectors, resulting in faster convergence and achieving the minimum loss function value. Per grid cell, YOLOv5 predicts numerous bounding boxes. However, only one box is necessary, which is decided by the intersection over union (IoU) with the ground truth with the highest value. During the training phase, the probability of objects in an image patch

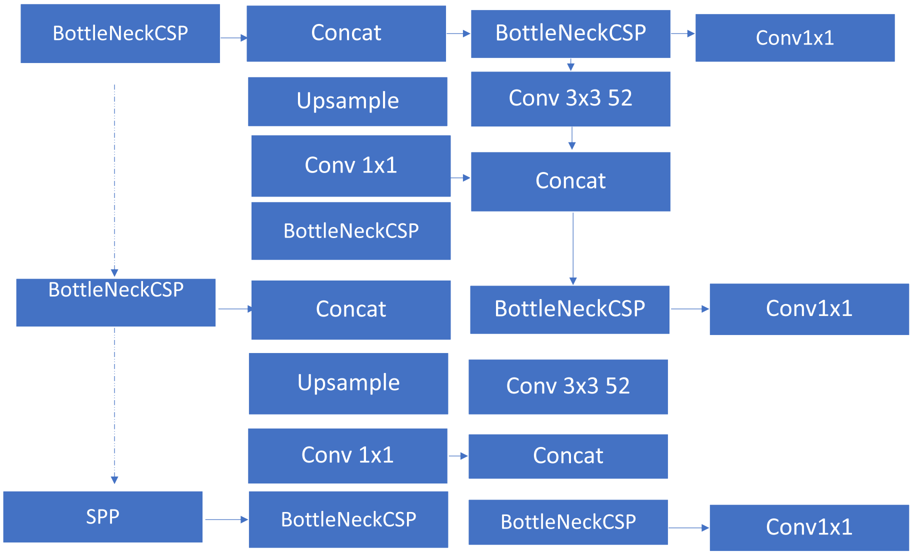


Fig. 6 Custom YOLOv5 network use to train, validate, and test the dataset (modified from Mishra et al. and Park et al. [63, 71])

region is modified to minimize the difference using offsetting. To quantify the loss for bounding box refinement, YOLOv5 adds sum-squared errors between model estimations and the actual value. Finally, the total loss function [54] given in Eq. 3 is used to translate the predicted bounding box to a ground truth bounding box using the geometrical coordinates and the confidence score [81].

$$\begin{aligned}\text{Total loss} &= \text{classification loss (CL)} + \text{localisation loss (LL)} \\ &\quad + \text{confidence loss (CL)}\end{aligned} \tag{3}$$

$$CL = \sum_{i=0}^{S^2} \coprod_{i}^{obj} (P_i(C) - \hat{P}_i(C))^2, \tag{4}$$

where $\coprod_i^{\mathrm{obj}}(P_i(C) = 1$ if an object is in cell i, otherwise $= 0$ and $\hat{P}_i(C)$ represents the conditional class probability for class C in cell i. The second term localization loss in Eq. 3 is defined as [54]:

$$\begin{aligned}LL = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \coprod_{ij}^{obj} [(X_i - \hat{X}_i)^2 + (Y_i - \hat{Y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \coprod_{ij}^{obj} [(\sqrt{W_i} - \sqrt{\hat{W}_i})^2 + (\sqrt{H_i} - \sqrt{\hat{H}_i})^2],\end{aligned} \tag{5}$$

where $\coprod_{ij}^{obj} = 1$ if the jth bounding box detects an object in cell i, otherwise $= 0$. λ_{coord} was set to 5 to compensate for the loss of the bounding box coordinates. This was done to place more focus on the correctness of the box. W_i and H_i are the width and height of the ground truth bounding box, while $\hat{W}_i$ and $\hat{H}_i$ are the width and height of the predicted bounding box by the model, respectively. Similarly, X_i, Y_i, and $\hat{X}_i$, $\hat{Y}_i$ are the coordinates of the center of the ground truth and predicted bounding boxes, respectively. The third term confidence loss in Eq. 3 is defined as [54]:

$$CL = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \coprod_{ij}^{obj} [(C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{i=0}^{B} \coprod_{ij}^{noobj} [(C_i - \hat{C}_i)^2], \tag{6}$$

where $\coprod_{ij}^{noobj}$ is the complement of $\coprod_{ij}^{obj}$. C_i is the confidence score decided before training the model, for the model to identify objects based on this threshold. $\hat{C}_i$ denotes the confidence score of the box j in cell i. $\lambda_{noobj} = 0.5$ to weight down the non-object loss to rectify the class imbalance problem, since majority of boxes do not contain any objects for classification. However, in YOLO, since loss function treats errors for small and large boxes equally, hence disproportionate boxes size in detecting defects might lead to incorrect predictions.

4.8 Hyperparameter tuning

The custom YOLOv5s network is trained over the 10291 images with an SGDM loss function (Eq. 3) by selecting a batch size of 16 and running over 100 epochs. The number of chosen epochs is optimized by considering detection accuracy and computation time. The first hyperparameter, momentum, is set to 0.937 and is used for updating parameters between iterations. The learning rate is determined by how frequently the weights get updated during training. Selecting a large learning rate can cause the model to converge quickly, requiring fewer epochs, while lower learning rates need more training epochs to get optimum weights. The learning rate used for this study is 0.01, and the weight decay is 0.0005, which is responsible for weight parameter reduction during backpropagation, thus adding a penalty component to the cost function.

The faster R-CNN is trained on the image dataset to test the model's performance and compare it with the custom YOLOv5 model. It is a state-of-the-art object detection model that gives high accuracy on a dataset in quasi-real-time. It uses a region proposal network for predicting object regions and classes, eliminating the need for the selective search algorithm, thus reducing the region proposal time drastically. The object detection method in general, comprises three steps, i.e., performing a forward pass, followed by calculating losses and updating the weights of the network. The main aim of the training is to minimize the loss and obtain an almost constant value. The accuracy of detection for the current model is determined using an IoU method, which measures the extent of overlap between the ground truth and the predicted bounding boxes. IoU is defined as the ratio of intersection area to union area. The threshold value of IoU is taken as 0.5 [89]. This means that only predicted bounding boxes with IoU ≥ 0.5 are considered as correct detection.

The mAP is used to determine the accuracy of detection. The faster R-CNN is implemented on Google Collab cloud GPU using TensorFlow 1.4, and the object detection API [90]. The faster R-CNN and RPN are trained using a momentum optimizer value of 0.9, and the first stage feature map stride for the sliding convolutional layer of RPN is 16. The training image dataset and the validation image dataset are used to calculate the gradient for backpropagation and to obtain an optimum number of iterations for training, respectively. Iterations affect precision and training time of the DL model. Fewer iterations reduce training time, but not training loss, leading to low detection and precision. Iterations help reduce and stabilise loss function.

However, too many iterations may lead to overfitting detection and waste time and computational resources. A set of alternative numbers of iterations in ascending order is utilized to identify the best number of iterations. The network is trained ten times, and the maximum detection accuracy is 85.04 % as measured by the mAP.

5 Results and discussion

The custom YOLOv5 model is trained, validated, and tested on a CH image dataset containing 10291 images with image sizes decreased to 416×416 pixels, with batch size set to 16 and the number of epochs to 100. The present study mainly focuses on developing a DL model that can detect multiple defects in a given CH structure. The number of epochs was set to 100 based on analysis and the trial-and-error method to optimize for ensuring a minimum loss function and time and computational efficiency. The performance metrics for the YOLOv5 model are obtained using the parameters mAP@0.5, precision, recall, and loss, namely, classification, box, and objectness, as a function of the number of epochs. The maximum mAP@0.5 obtained is 93.7% for four types of defects, i.e., 98.9% for cracks, 85.3% for discoloration, 96.4% for exposed bricks, and 94.2% for spalling, as shown in Table 1. The faster R-CNN could detect the defects more accurately for some instances, but it gives less overall accuracy when looked at for multiple defects. The precision and recall are 85.9 and 91.8%, respectively. The confidence score is selected as 0.5, which means the model identifies defects with this as the minimum accuracy. The graphs and model performance over test images contained in the 10% split of the original dataset are shown in Figs. 7, 8, 9.

5.1 Damage detection results and performance metrics from the custom YOLOv5 model

An image classifier is trained on the final dataset of CH photos, and it is then used to detect defects in the dataset's images of CH sites. Examples of the classifier's output are presented in Figs. 7 and 8. Each detected bounding box has a detection confidence level indicated in the upper-right corner. The confidence level for detecting several flaws is within an acceptable range (56–97%). For the detection of defects, a variety of site conditions are analyzed. In Fig. 7a-d exposed bricks are detected with a confidence level in the range 75–97%. In the same image, Fig. 7e illustrates two types of defects: discoloration and exposed brickwork. Discoloration detected within the area of the exposed brick with a confidence level of 74% is highly accurate. Figure 7f illustrates two classes of defects, namely, discoloration and exposed brick, which are detected satisfactorily. In Fig. 8g, discoloration is detected with an accuracy of 71–72%; in Fig. 8h-j, exposed brick is detected accurately. Figure 8k and l detects three and two types of defects, respectively, and these show that the model can detect instances of crack over lantern-shaped structures over the dome.

Table 1 Performance metrics corresponding to each class of defects

Defects	mAP YOLOv5	mAP Faster R-CNN	Precision YOLOv5	Recall YOLOv5
Spalling	94.2	87.9	82.5	94.3
Discoloration	85.3	89.9	79.8	80.4
Exposed bricks	96.4	96.9	90.6	92.6
Cracks	98.9	65.76	90.8	99.1
All	93.7	85.1	85.9	91.8

The results from the custom YOLOv5 model are plotted in Fig. 9. The mAP for the model is 93.7% (Fig. 9c), which is achieved at approximately 50 epochs, and then the graph reaches saturation at 100 epochs. The mAP the model is 85.9%, as shown in Fig. 9b, and the maximum recall is 91.8% (Fig. 9a). The precision in Fig. 9b passes through a dip at approximately 30 epochs, mainly because the model is learning over a new dataset and determining the number of TPs. The recall from Fig. 9a increases continuously and indicates that the number of correct predictions gradually increases with the number of epochs, reaching its maximum value at approximately 50 epochs. The loss function, shown in Fig. 9d-f, indicates the errors with which the model detects an object, gradually decreasing with each training epoch.

5.2 Damage detection results and performance metrics from the faster R-CNN model

As shown in Fig. 10, the faster R-CNN can successfully detect multiple defects in a single test image. Fig. 10a–i shows detection cases of discoloration and exposed brick in the range 46–91%. Figure 10d detects exposed brick with multiple bounding boxes. This is because the ground truth bounding boxes used for training are similar to it, and they provide better localization results. Figure 10d shows an instance of crack over the lantern located at the top of the dome; the model was unable to detect the crack but detected spalling with 89% accuracy at the bottom of dome, but the YOLOv5 model detected it successfully. The faster R-CNN gives a particular confidence score for each of the bounding boxes that captures the defect. No single bounding box has two confidence scores (multiple accuracy), and the reason for having smaller boxes in the results is that the labeling was done in the same way for better localization of the defects. However, the object detection model can have partially overlapping bounding boxes, as for some cases, the rectangular area can have multiple defects

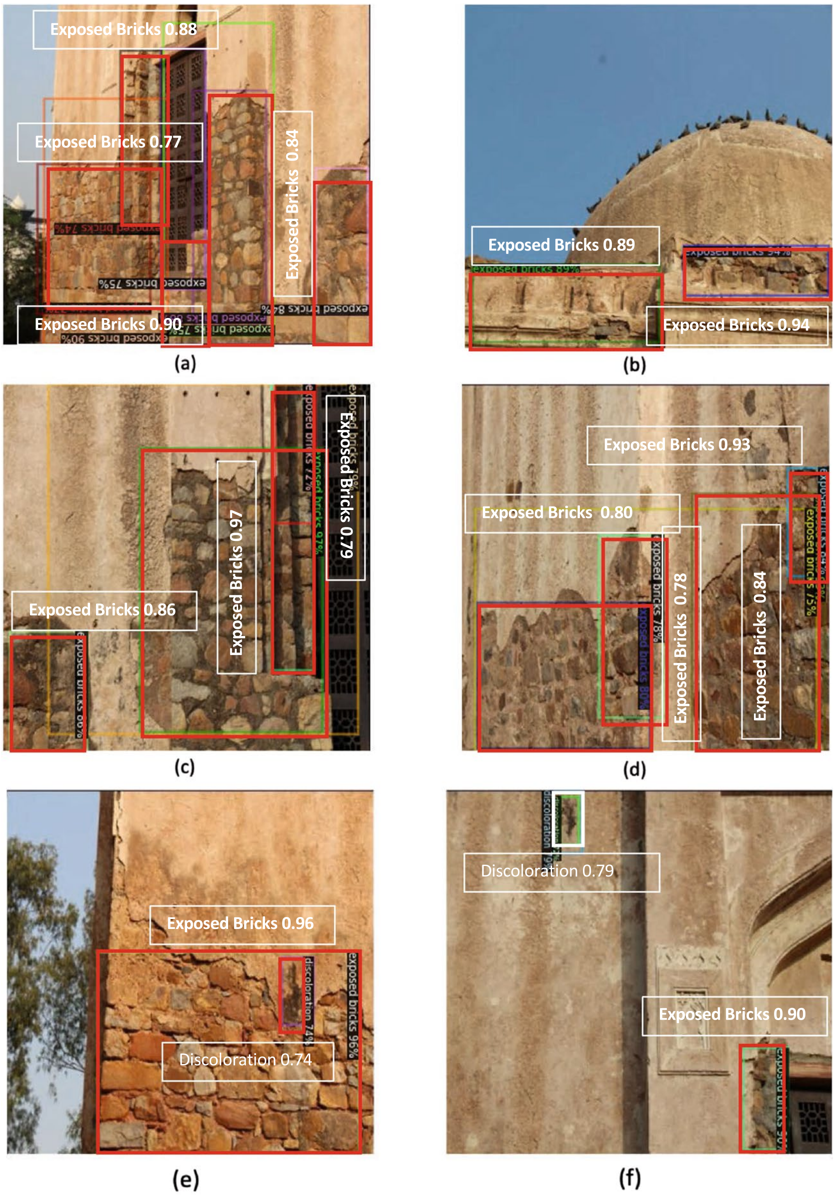


Fig. 7 Defect detection results from the custom YOLOv5 defect detection model (Some labels have been modified for enhanced visibility)

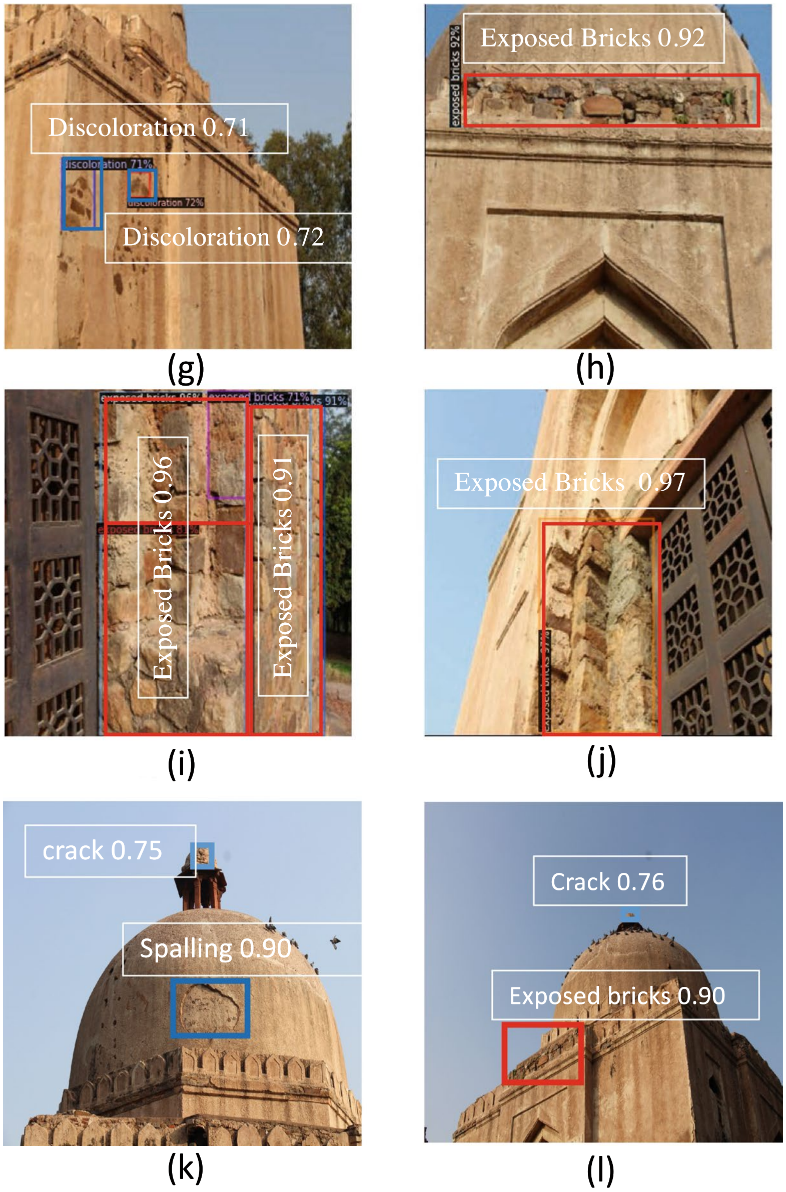
within the same region such as discoloration and exposed bricks shown in Fig. 10g.

The mAP achieved from training the faster R-CNN model as shown in Fig 11 is 85.04%, which is achieved after training the model for 8000 iterations, even though the maximum value is reached at approximately 6500 epochs. The class-wise mAP for exposed bricks is 96.9%, 89.9% for discoloration, 87.9% for spalling and 65.76% for cracks. The model gives sufficient accuracy, but adding to the dataset will ensure better accuracy.

5.3 Comparison of the custom YOLOv5 network and faster R-CNN

To compare the performance of the proposed YOLOv5, a ResNet 101 architecture-based faster R-CNN is used and trained on the same dataset. Cracks, spalling, exposed brick, and discoloration are detected successfully. The mAPs for faster R-CNN and proposed YOLOv5 are 85.04% and 93.7%, respectively. The results of both DL-models are comparable in terms of defect detection accuracy. The faster R-CNN requires more than four times the training time of the YOLOv5s model, i.e., 8 h 38 min versus 1 h 53 min. Thus, the proposed YOLOv5 is superior, with fewer false detection and considerably faster training and inference speed. Some instances of model training, with custom YOLOv5s, custom YOLOv5l and faster R-CNN are shown in Table 2. The YOLOv5s gives comparable results for the same dataset at approximately one-third of the time taken by the faster R-CNN.

Fig. 8 Defect detection results from the custom YOLOv5 defect detection model (Some labels have been modified for enhanced visibility)



5.4 Comparison of proposed model with other automatic visual inspection systems

This study aimed to determine the lack of research in defect detection systems and how AI-based visual inspection systems can be used to move in this direction. The results obtained for multi-class defects are often more challenging and less accurate compared to instances where only one category of defects, such as cracks, needs to be identified. Mansuri and Patel [31] developed an automatic web-based visual inspection system based on faster R-CNN inception v2 architecture that can detect three classes of defects with an accuracy of 91.5%. Chen et al. [51] used CNN to detect only cracked and uncracked concrete spaces with an accuracy of 99.71%. Wang et al. [1] detected two classes of defects and achieved an accuracy of 95%, while Cha et al. [52] detected five damage types with 87.8% accuracy. Cosovic and Jankovic [91] applied CNN for categorizing CH images into 10 categories with an accuracy of up to 90%. Although their work didn't directly identify damages, the identification of various components/objects [92] in the CH technique can be extended to damage detection in CH buildings. Wang et al. [93] GreatWatcher platform based on R-CNN DL framework gave 78.2% accuracy on a trained dataset on a small image sample of 610 images. Another study by the same research group of Wang et al. [1] used faster R-CNN and reported average precision of 0.999 and 0.900 for efflorescence and spalling damage in old masonry construction. In this study, we successfully developed a model that can detect four types of defects,

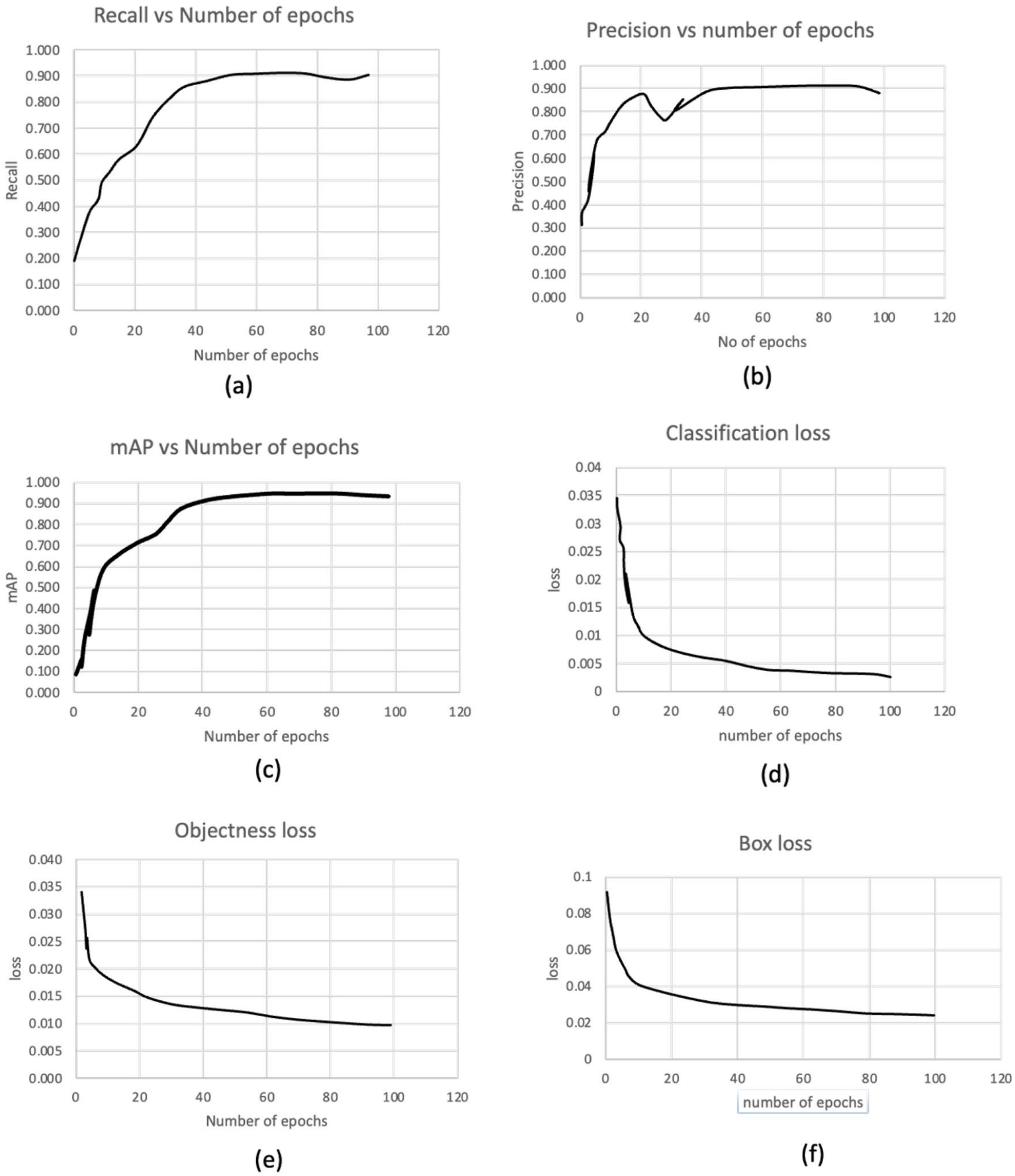


Fig. 9 Performance metrics of custom YOLOv5 defect detection model **a** recall, **b** precision, **c** mAP, **d** classification loss, **e** objectness loss, **f** box loss

namely, spalling, discoloration, exposed brick, and cracks, with an mAP of 93.7%. Hence, in comparison with other automatic damage detection DL techniques, the proposed model reports good performance and, at the same time, potential for real-time detection of CH defects. Kwon and Yu [94] classified stone-related damages typical of CH sites into four types (i.e., crack, material loss, detachment of material, biological colonization) based on the Faster CNN algorithm and achieved a confidence score of 94.6%. Samhouri et al. [95] employed CNN for detecting surface damages in architectural CH buildings in Jordan for four defects (erosion, material loss, color change of the stone, and sabotage issues in CH) with an accuracy of 95-96%. The proposed approach can be used similarly to contemporary methods of automatic visual inspections of historic buildings.

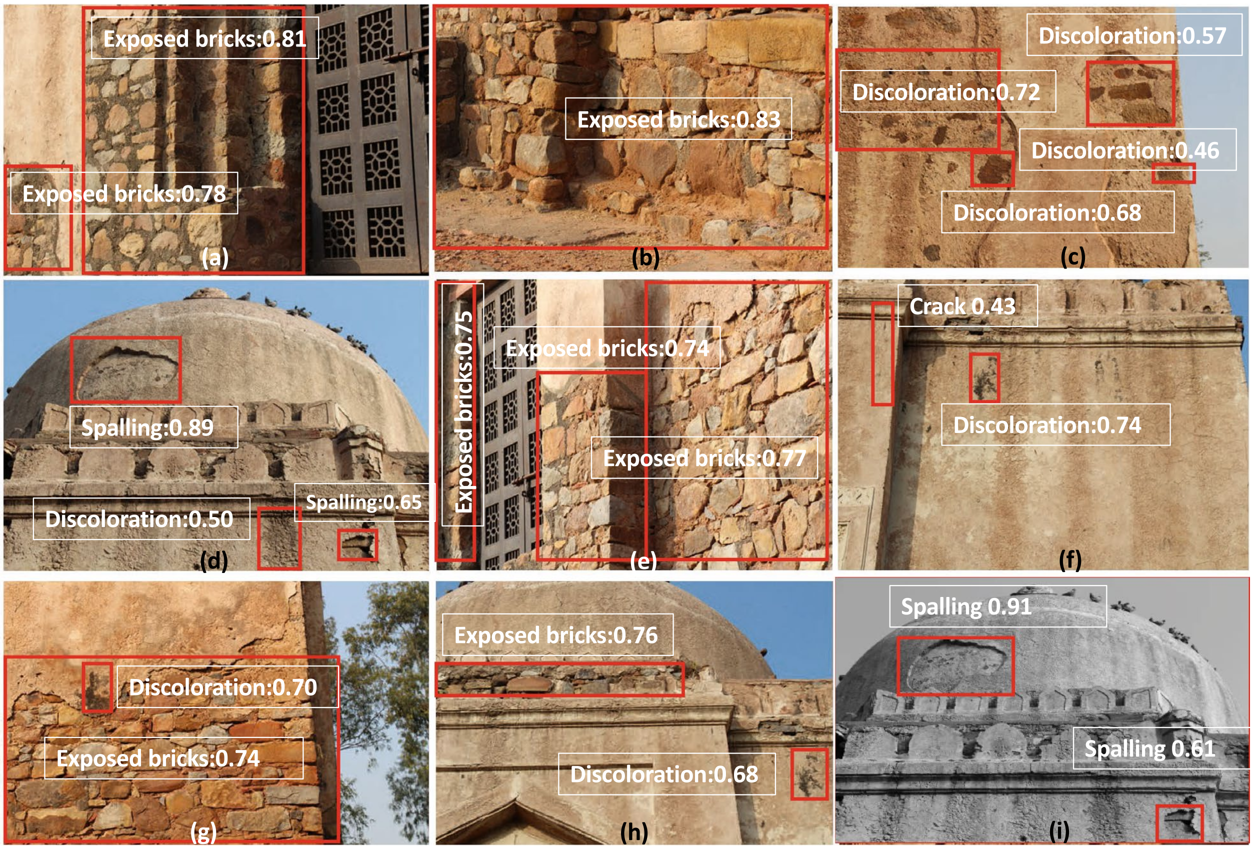


Fig. 10 Defect detection results from the faster R-CNN comparison model

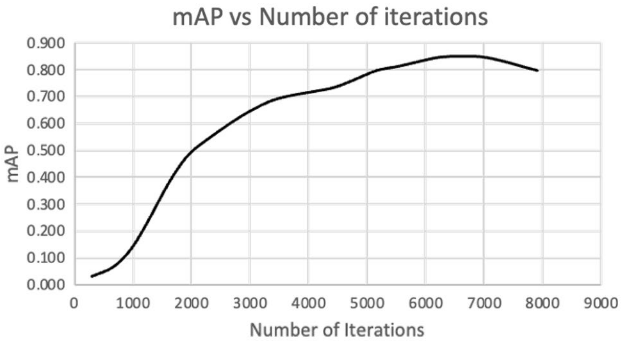


Fig. 11 Maximum average precision from the faster R-CNN model

Table 2 Model training time, number of defects taken, image dataset, epochs, and mAP for various DL models

Model	No. of defects	No. of images	Epochs	mAP	Time
Custom YOLOv5s	3	820	100	75.7%	48 mins
Custom YOLOv5s	4	4420	85	76.8%	3 h 14 mins
Custom YOLOv5l	2	559	100	77.9%	5 h 15 mins
Custom YOLOv5l	2	2372	150	79.2%	6 h 36 mins
Custom YOLOv5s	4	4451	100	85.7%	4 h 42 mins
Custom YOLOv5s	4	10291	50	93.7%	1 h 53 mins
Faster R-CNN	4	10291	8000	85.04%	8 h 38 mins

6 Conclusions, limitations, and future scope

This study proposes an automatic heritage structure defect detection model to aid in ensuring the sustainability of CH structures; it develops an automated visual inspection system that can accelerate the preservation and maintenance processes. The YOLOv5 DL model is used to produce automatic defect detection. The dataset has 10291 images with four types of surface defects: spalling, exposed brick, discoloration, and cracks. The Dadi-Poti tomb dataset is annotated and used as an image dataset. With the highest detection accuracy (mAP) of 0.937, the system can successfully detect four types of defects in CH structures. The model's performance is evaluated in various settings, including background noise and images taken from different angles. Successful application of this case study can help identify structural anomalies in need of urgent repair, thereby facilitating an improved civil infrastructure monitoring system. YOLOv5 can be enhanced by adding more photographs to the database and modifying the design.

The originality and contribution of this study are the application of the DL model to detect damages using the Dadi-Poti tomb as a case study. The custom YOLOv5 automatic inspection system is tested and validated by comparing it with a ResNet 101-based faster R-CNN model, and it can be used by conservation authorities to conduct regular inspections at a lower cost; it will allow them to make more timely decisions about repair and maintenance work to be carried out in the built environment. The proposed YOLO model gives better accuracy for classifying multiple defects in CH and almost a quarter of the training time when compared with its counterpart, R-CNN. This reduction in training time is helpful for practical purposes, thus saving computational costs and enabling faster model deployment. A random set of test photos is used to test the model. We have attached a small video of 20 seconds in supplementary material that could be the prototype for future works. If the images/video is captured using a drone, then YOLO can give damages in real-time (as video is just the number of frames in a second). The findings of this study can encourage the use of automatic systems instead of manual inspection to save time and money in terms of labor costs. This method is beneficial to inspection engineers, material scientists, and the overall heritage conservation community; it can also boost the development of new damage detection techniques. The suggested automated inspection can help conservation agencies manage their finances. Furthermore, the paper deals with the SHM of CH. Still, the framework can be extended to other areas of CH preservation, such as the classification of architectural elements within heritage buildings, identifying disaster-affected CH, artwork identification, and image reconstruction in CH.

The proposed YOLO model is confined to detecting four types of typical surface flaws in CH structures in this research work. More categories of defects, such as efflorescence, seepage, dust deposition, fungal growth, and missing components, can be considered in future studies, and a comparable model can be developed using the proposed method. Threats to the validity of the DL models should be taken into account, such as the quality of the image dataset, accurate labelling of various defects and sufficient images of each defect, in particular for the YOLO model detection of minor defects that appear in groups of defects. Future research can be focused on severity assessment as well as defect quantification. The current research utilizes image datasets, but future works include running the YOLO model on the input obtained from UAVs and computer webcams. Moreover, the model can be further developed to achieve higher performance with a smaller dataset and less computation time.

The present stage of research is based on detecting defects that are present in the collected images. Still, this automatic detection of defects does not tell about the spatial location of defects on the CH structure itself. Therefore, future studies can develop a model to locate the object and geo-tagging it at the exact position of the CH structure, taking the ground frame of reference to tell the precise location coordinates of the defect over the structure in 3D platforms.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s13349-022-00643-8.

Data availability The Mendeley database (https://data.mendeley.com/datasets/gnyzwrz4gt/1) has some of the image data collected by authors, Marco B and Michela C, that was used for the YOLOv5 models. The author can provide the Python code for replication of results upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Wang N, Zhao X, Zhao P, Zhang Y, Zou Z, Jinping O (2019) Automatic damage detection of historic masonry buildings based on mobile deep learning. Autom Constr 103:53–66
2. Agdas D, Rice JA, Martinez JR, Lasa IR (2016) Comparison of visual inspection and structural-health monitoring as bridge condition assessment methods. J Perform Constr Facil 30(3):04015049
3. Georgopoulos GD, Telioni EC, Tsontzou A (2016) The contribution of laser scanning technology in the estimation of ancient greek monuments' deformations. Surv Rev 48(349):303–308
4. Costanzo A, Minasi M, Casula G, Musacchio M, Buongiorno MF (2014) Combined use of terrestrial laser scanning and ir thermography applied to a historical building. Sensors 15(1):194–213
5. Armesto-González J, Riveiro-Rodríguez B, González-Aguilera D, Rivas-Brea MT (2010) Terrestrial laser scanning intensity data applied to damage detection for historical buildings. J Archaeol Sci 37(12):3037–3047
6. Yuan L, Guo J, Wang Q (2020) Automatic classification of common building materials from 3d terrestrial laser scan data. Autom Constr 110:103017
7. Mishra M, Lourenço PB, Ramana GV (2022) Structural health monitoring of civil engineering structures by using the internet of things: a review. J Build Eng 48:103954. https://doi.org/10.1016/j.jobe.2021.103954
8. Ramos LF, Miranda T, Mishra M, Fernandes FM, Manning E (2015) A bayesian approach for ndt data fusion: the saint torcato church case study. Eng Struct 84:120–129
9. Nghiem H-L, Heib M, Emeriault F (2015) Method based on digital image correlation for damage assessment in masonry structures. Eng Struct 86:1–15

10. Torres B, Varona FB, Baeza FJ, Bru D, Ivorra S (2020) Study on retrofitted masonry elements under shear using digital image correlation. Sensors 20(7):2122
11. Rezaie A, Achanta R, Godio M, Beyer K (2020) Comparison of crack segmentation using digital image correlation measurements and deep learning. Construct Build Mater 261:120474
12. Galantucci RA, Fatiguso F (2019) Advanced damage detection techniques in historical buildings using digital photogrammetry and 3d surface anlysis. J Cult Herit 36:51–62
13. Kim B, Cho S (2019) Image-based concrete crack assessment using mask and region-based convolutional neural network. Struct Control Health Monit 26(8):e2381
14. Perumal R, Venkatachalam SB (2021) Non invasive detection of moss and crack in monuments using image processing techniques. J Ambient Intell Humaniz Comput 12(5):5277–5285
15. Adamopoulos E (2021) Learning-based classification of multispectral images for deterioration mapping of historic structures. J Build Pathol Rehabilitation 6(1):1–15
16. Newman C, Edwards D, Martek I, Lai J, Thwala WD, Rillie I (2021) Industry 4.0 deployment in the construction industry: a bibliometric literature review and UK-based case study. Smart Sustain Built Environ 10(4):557–580. https://doi.org/10.1108/SASBE-02-2020-0016
17. Rahimian FP, Goulding JS, Abrishami S, Seyedzadeh S, Elghaish F (2021) Industry 4.0 solutions for building design and construction: a paradigm of new opportunities. Routledge, p 420. eBook ISBN 9781003106944. https://doi.org/10.1201/9781003106944
18. Prieto AJ, Ortiz R, Macías-Bernal JM, Chávez MJ, Ortiz Pi (2019) Artificial intelligence applied to the preventive conservation of heritage buildings. In Science and Digital Technology for Cultural Heritage. CRC Press pages 245–249
19. Bienvenido-Huertas D, Nieto-Julián JE, Moyano JJ, Macías-Bernal JM, Castro J (2019) Implementing artificial intelligence in h-bim using the J48 algorithm to manage historic buildings. Int J Archit Herit 14(8):1148–1160. https://doi.org/10.1080/15583058.2019.1589602
20. Sánchez-Aparicio LJ, Masciotta M-G, García-Alvarez J, Ramos LF, Oliveira DV, Martín-Jiménez JAn, González-Aguilera D, Monteiro P (2020) Web-gis approach to preventive conservation of heritage buildings. Autom Construct 118:103304
21. Sony S, Dunphy K, Sadhu A, Capretz M (2021) A systematic review of convolutional neural network-based structural condition assessment techniques. Eng Struct 226:111347
22. Nazarian E, Taylor T, Weifeng T, Ansari F (2018) Machine-learning-based approach for post event assessment of damage in a turn-of-the-century building structure. J Civ Struct Health Monit 8(2):237–251
23. Mishra M (2021) Machine learning techniques for structural health monitoring of heritage buildings: a state-of-the-art review and case studies. J Cult Herit 47:227–245
24. Zou Z, Zhao P, Zhao X (2021) Automatic segmentation, inpainting, and classification of defective patterns on ancient architecture using multiple deep learning algorithms. Struct Control Health Monit 28(7):e2742
25. Trier ØD, Cowley DC, Waldeland AU (2019) Using deep neural networks on airborne laser scanning data: results from a case study of semi-automatic mapping of archaeological topography on arran, scotland. Archaeol Prospect 26(2):165–175
26. Mishra M, Bhatia AS, Maity D (2020) Predicting the compressive strength of unreinforced brick masonry using machine learning techniques validated on a case study of a museum through non-destructive testing. J Civil Struct Health Monit 10(3):389–403
27. Mansuri LE, Patel DA (2022) Development of automated web-based condition survey system for heritage monuments using deep learning. In: Belayutham S, Che Ibrahim CKI, Alisibramulisi A, Mansor H, Billah M (eds) Proceedings of the 5th International Conference on Sustainable Civil Engineering Structures and Construction Materials. Lecture Notes in Civil Engineering, vol 215. Springer, Singapore. https://doi.org/10.1007/978-981-16-7924-7_76
28. Yao G, Sun Y, Wong M, Lv X (2021) A real-time detection method for concrete surface cracks based on improved yolov4. Symmetry 13(9):1716
29. Narazaki Yasutaka, Hoskere Vedhus, Yoshida Koji, Spencer Billie F, Fujino Yozo (2021) Synthetic environments for vision-based structural condition assessment of japanese high-speed railway viaducts. Mech Syst Signal Process 160:107850
30. Hoskere V, Narazaki Y, Spencer BF Jr (2022) Physics-based graphics models in 3d synthetic environments as autonomous vision-based inspection testbeds. Sensors 22(2):532
31. Mansuri LE, Patel DA (2021) Artificial intelligence-based automatic visual inspection system for built heritage. Smart Sustain Built Environ. https://doi.org/10.1108/SASBE-09-2020-0139
32. Mansuri LE, Patel DA (2022) Artificial intelligence for heritage conservation: a case study of automatic visual inspection system. In: Li RYM, Chau KW, Ho DCW (eds) Current state of art in artificial intelligence and ubiquitous cities. Springer, Singapore. https://doi.org/10.1007/978-981-19-0737-1_1
33. LabelImg Tzutalin (2015) Git code. https://github.com/tzutalin/labelImg. Accessed 10 Mar 2022
34. Chaiyasarn K, Sharma M, Ali L, Khan W, Poovarodom N (2018) Crack detection in historical structures based on convolutional neural network. GEOMATE J 15(51):240–251
35. Dais D, Bal IE, Smyrou E, Sarhosis V (2021) Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. Autom Construct 125:103606
36. Wang N, Zhao Q, Li S, Zhao X, Zhao P (2018) Damage classification for masonry historic structures using convolutional neural networks based on still images. Comput-Aided Civil Infrastruct Eng 33(12):1073–1089
37. Wang N, Zhao X, Zou Z, Zhao P, Qi F (2020) Autonomous damage segmentation and measurement of glazed tiles in historic buildings via deep learning. Comput-Aided Civil Infrastruct Eng 35(3):277–291
38. Guo J, Wang Q, Li Y (2021) Evaluation-oriented façade defects detection using rule-based deep learning method. Autom Construct 131:103910
39. Monna F, Rolland T, Denaire A, Navarro N, Granjon L, Barbé R, Chateau-Smith C (2021) Deep learning to detect built cultural heritage from satellite imagery.-spatial distribution and size of vernacular houses in sumba, indonesia. J Cult Herit 52:171–183
40. Mondal TG, Jahanshahi MR, R-TW, Zheng YW (2020) Deep learning-based multi-class damage detection for autonomous post-disaster reconnaissance. Struct Control Health Monit 27(4):e2507
41. Sharma E, Agrawal P, Verma NK (2019) Detection of dust deposition using convolutional neural network for heritage images. In Computational Intelligence: Theories, Applications and Future Directions-Volume II, pages 347–359. Springer
42. Masrour T, El Hassani I, Bouchama MS (2020) Deep convolutional neural networks with transfer learning for old buildings pathologies automatic detection. In: Ezziyyani M (eds) Advanced Intelligent Systems for Sustainable Development (AI2SD'2019). AI2SD 2019. Advances in Intelligent Systems and Computing, vol 1104. Springer, Cham. https://doi.org/10.1007/978-3-030-36671-1_18
43. Zou Z, Zhao X, Zhao P, Qi F, Wang N (2019) Cnn-based statistics and location estimation of missing components in routine inspection of historic buildings. J Cult Herit 38:221–230

44. Masrour T, Hassani IE, Bouchama MS (2019) Deep convolutional neural networks with transfer learning for old buildings pathologies automatic detection. In International Conference on Advanced Intelligent Systems for Sustainable Development, pages 204–216. Springer
45. Dung CV et al (2019) Autonomous concrete crack detection using deep fully convolutional neural network. Autom Construct 99:52–58
46. Yang X, Li H, Yantao Y, Luo X, Huang T, Yang X (2018) Automatic pixel-level crack detection and measurement using fully convolutional network. Comput-Aided Civil Infrastruct Eng 33(12):1090–1109
47. Azimi M, Eslamlou AD, Pekcan G (2020) Data-driven structural health monitoring and damage detection through deep learning: state-of-the-art review. Sensors 20(10):2778
48. Han Q, Pan Y, Yang D, Ying X (2022) CNN-based bolt loosening identification framework for prefabricated large-span spatial structures. J Civil Struct Health Monitor 12:517–536. https://doi.org/10.1007/s13349-022-00561-9
49. Zhou Q, Ding S, Qing G, Jingbo H (2022) Uav vision detection method for crane surface cracks based on faster r-cnn and image segmentation. J Civil Struct Health Monitor 1–11
50. Kung R-Y, Pan N-H, Wang CCN, Lee P-C (2021) Application of deep learning and unmanned aerial vehicle on building maintenance. Adv Civil Eng 2021:5598690. https://doi.org/10.1155/2021/5598690
51. Chen K, Yadav A, Khan A, Meng Y, Zhu K (2019) Improved crack detection and recognition based on convolutional neural network. Model Simul Eng 2019:8796743. https://doi.org/10.1155/2019/8796743
52. Cha Y-J, Choi W, Suh G, Mahmoudkhani S, Büyüköztürk O (2018) Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput-Aided Civil Infrastruct Eng 33(9):731–747
53. Liu Z, Cao Y, Wang Y, Wang W (2019) Computer vision-based concrete crack detection using u-net fully convolutional networks. Autom Construct 104:129–139
54. Deng J, Ye L, Lee VC-S (2021) Imaging-based crack detection on concrete surfaces using you only look once network. Struct Health Monit 20(2):484–499
55. Feng C, Zhang H, Li Y, Wang S, Wang H (2021) Efficient real-time defect detection for spillway tunnel using deep learning. J Real-Time Image Process 18(6):2377–2387
56. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition 779–788
57. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition 7263–7271
58. Park SE, Eem S-H, Jeon H (2020) Concrete crack detection and quantification using deep learning and structured light. Construct Build Mater 252:119096
59. Jingwei LX, Yang SL, Wang X, Luo S, Lee VC-S, Ding L (2020) Automated pavement crack detection and segmentation based on two-step convolutional neural network. Comput-Aided Civil Infrastruct Eng 35(11):1291–1305
60. Jiang S, Zhang J (2020) Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system. Comput-Aided Civil Infrastruct Eng 35(6):549–564
61. Teng S, Liu Z, Chen G, Cheng L (2021) Concrete crack detection based on well-known feature extractor model and the yolo_v2 network. Appl Sci 11(2):813
62. Li S, Xingyu G, Xiangrong X, Dawei X, Zhang T, Liu Z, Dong Q (2021) Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm. Construct Build Mater 273:121949
63. Mishra M, Jain V, Singh SK, Maity D (2022) Two-stage method based on the you only look once framework and image segmentation for crack detection in concrete structures. Archit Struct Construct 2(1):1–18
64. Maeda H, Sekimoto Y, Seto T, Kashiyama T, Omata H (2018) Road damage detection and classification using deep neural networks with smartphone images. Comput-Aided Civil Infrastruct Eng 33(12):1127–1141
65. Yuchuan D, Pan N, Zihao X, Fuwen DY, Shen, Hua K (2020) Pavement distress detection and classification based on YOLO network. Int J Pavement Eng 22(13):1659–1672
66. Fu-Jun D, Jiao S-J (2022) Improvement of lightweight convolutional neural network model based on yolo algorithm and its research in pavement defect detection. Sensors 22(9):3537
67. Liu Z, Wenxiu W, Xingyu G, Li S, Wang L, Zhang T (2021) Application of combining yolo models and 3d gpr images in road detection and maintenance. Remote Sens 13(6):1081
68. Zhou S, Yufeng BX, Wei JL, Ye Z, Li F, Yuchuan D (2021) Automated detection and classification of spilled loads on freeways based on improved yolo network. Mach Vis Appl 32(2):1–12
69. Hou X, Zeng Y, Xue J (2020) Detecting structural components of building engineering based on deep-learning method. J Construct Eng Manag 146(2):04019097
70. Ukhwah EN, Yuniarno Eko M, Suprapto YK (2019) Asphalt pavement pothole detection using deep learning method based on yolo neural network. In 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), pages 35–40. IEEE
71. Park S-S, Tran V-T, Lee D-E (2021) Application of various yolo models for computer vision-based real-time pothole detection. Appl Sci 11(23):11229
72. Ge Liangfu, Dan Danhui, Li Hui (2020) An accurate and robust monitoring method of full-bridge traffic load distribution based on yolo-v3 machine vision. Struct Control Health Monit 27(12):e2636
73. RongXin W et al. (2019) Research on safety helmet wearing yolo-v3 detection technology improvement in mine environment. In Journal of Physics: Conference Series, volume 1345(4), page 042045. IOP Publishing
74. Hu J, Gao X, Wu H, Gao S (2019) Detection of workers without the helments in videos based on yolo v3. In 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), pages 1–4. IEEE
75. Li Yang, Chen Jun (2022) Computer vision-based counting model for dense steel pipe on construction sites. J Construct Eng Manag 148(1):04021178
76. Park Sehwan, Kim Jinpyung, Jeon Kyoyoung, Kim Junkyeong, Park Seunghee (2011) Improvement of gpr-based rebar diameter estimation using yolo-v3. Remote Sens 13(10):2021b
77. Yang Xinyue, Gao Yuqing, Fang Cheng, Zheng Yue, Wang Wei (2022) Deep learning-based bolt loosening detection for wind turbine towers. Struct Control Health Monit 29(6):e2943
78. Zhao Y, Deng X, Lai H (2020) A yolo-based method to recognize structural components from 2d drawings. In Construction Research Congress 2020: Computer Applications, pages 753–762. American Society of Civil Engineers Reston, VA
79. Idjaton K, Desquesnes X, Treuillet S, Brunetaud X (2022) Transformers with yolo network for damage detection in limestone wall images. In International Conference on Image Analysis and Processing, pages 302–313. Springer
80. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
81. Malta A, Mendes M, Farinha T (2021) Augmented reality maintenance assistant using yolov5. Appl Sci 11(11):4758

82. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28:91–99
83. Girshick R (2015) Fast r-cnn. In Proceedings of the IEEE international conference on computer vision 1440–1448
84. Cha Y-J, Choi W, Büyüköztürk O(2017) Deep learning-based crack damage detection using convolutional neural networks. Comput-Aided Civil Infrastruct Eng 32(5):361–378
85. Kumar SS, Wang M, Abraham DM, Jahanshahi MR, Iseley T, Cheng JCP (2020) Deep learning-based automated detection of sewer defects in cctv videos. J Comput Civil Eng 34(1):04019047
86. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In European conference on computer vision, pages 740–755. Springer
87. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al. (2016) {TensorFlow}: A system for {Large-Scale} machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16), pages 265–283
88. Chollet F (2015) keras, GitHub. https://github.com/fchollet/keras
89. Everingham M, Gool L Van, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
90. Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I et al (2017) Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7310–7311. Tensorflow object detection api. Code: https://github.com/tensorflow/models/tree/master/research/object_detection
91. Ćosović M, Janković R (2020) Cnn classification of the cultural heritage images. In 2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH), pages 1–6. IEEE
92. Trier ØD, Reksten JH, Løseth K (2021) Automated mapping of cultural heritage in norway from airborne lidar data using faster r-cnn. Int J Appl Earth Obs Geoinform 95:102241
93. Wang N, Zhao X, Wang L, Zou Z (2019) Novel system for rapid investigation and damage detection in cultural heritage conservation based on deep learning. J Infrastruct Syst 25(3):04019020
94. Kwon D, Jeongmin Y (2019) Automatic damage detection of stone cultural property based on deep learning algorithm. Int Arch Photogramm Remote Sens Spat Inf Sci 42:639–643
95. Samhouri M, Al-Arabiat L, Al-Atrash F (2022) Prediction and measurement of damage to architectural heritages facades using convolutional neural networks. Neural Comput Appl 34:18125–18141. https://doi.org/10.1007/s00521-022-07461-5
96. Mishra M, Ramana GV (2022) Data for Dadi-Poti (Cultural Heritage) - Sample Image Dataset used for Automatic Visual Inspection System, Mendeley Data, V2. https://doi.org/10.17632/gnyzwrz4gt.2

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