



Modern Crack Detection for Bridge Infrastructure Maintenance Using Machine Learning

Hafiz Suliman Munawar¹ · Ahmed W. A. Hammad¹ · S. Travis Waller² · Md Rafiqul Islam³

Received: 8 July 2022 / Accepted: 11 September 2022 / Published online: 28 September 2022
© The Author(s) 2022

Abstract

Manual investigation of damages incurred to infrastructure is a challenging process, in that it is not only labour-intensive and expensive but also inefficient and error-prone. To automate the process, a method that is based on computer vision for automatically detecting cracks from 2D images is a viable option. Amongst the different methods of deep learning that are commonly used, the convolutional neural network (CNNs) is one that provides the opportunity for end-to-end mapping/learning of image features instead of using the manual suboptimal image feature extraction. Specifically, CNNs do not require human supervision and are more suitable to be used for indoor and outdoor applications requiring image feature extraction and are less influenced by internal and external noise. Additionally, the CNN's are also computationally efficient since they are based on special convolution layers and pooling operations that enable the full execution of CNN frameworks on several hardware devices. Keeping this in mind, we propose a deep CNN framework that is based on 10 different convolution layers along with a cycle GAN (Generative Adversarial Network) for predicting the crack segmentation pixel by pixel in an end-to-end manner. The methods proposed here include the Deeply Supervised Nets (DSN) and Fully Convolutional Networks (FCN). The use of DSN enables integrated feature supervision for each stage of convolution. Furthermore, the model has been designed intricately for learning and aggregating multi-level and multiscale features while moving from the lower to higher convolutional layers through training. Hence, the architecture in use here is unique from the ones in practice which just use the final convolution layer. In addition, to further refine the predicted results, we have used a guided filter and CRFs (Conditional Random Fields) based methods. The verification step for the proposed framework was carried out with a set of 537 images. The deep hierarchical CNN framework of 10 convolutional layers and the Guided filtering achieved high-tech and advanced performance on the acquired dataset, showing higher F-score, Recall and Precision values of 0.870, 0.861, and 0.881 respectively, as compared to the traditional methods such as SegNet, Crack-BN, and Crack-GF.

Keywords Damage detection · Crack detection · Flood disaster · Machine learning · Convolutional neural networks · Cycle generative adversarial network (CycleGAN) · Unmanned aerial vehicles

✉ Hafiz Suliman Munawar
h.munawar@unsw.edu.au

Ahmed W. A. Hammad
a.hammad@unsw.edu.au

S. Travis Waller
steventravis.waller@tu-dresden.de

Md Rafiqul Islam
rafiqulislam.cse24@gmail.com

- ¹ School of the Built Environment, University of New South Wales, Sydney NSW2052, Australia
- ² Chair of Transport Modelling and Simulation “Friedrich List”, Faculty of Transport and Traffic Sciences, Technische Universität Dresden, Dresden, Germany
- ³ Data Science Institute (DSI), University of Technology Sydney (UTS), Sydney, Australia

1 Introduction

Civil infrastructure including bridges, roads and tunnels remain vulnerable to deterioration due to the occurrence of disasters, along with cyclical loading and harsh environmental factors [1]. Timely detection and simultaneous maintenance of civil infrastructure is an indispensable way to ensure human safety and reduction in the costs associated with infrastructural damages [2, 3]. As reported in the literature, structural deficiency along with aging and respective failure, have been associated with the damages which ultimately lead to the collapse of the majority of bridges (~46%) [4]. Effective detection and maintenance of

the infrastructural health that is exposed to various types of damages in the form of corrosion and cracks are thus important [5].

One alarming and frequently occurring infrastructural damage is the appearance of cracks. Generally, cracks initiate on the surfaces of the concrete structures mainly due to stress, fatigue, cyclic loading, poor construction, deterioration/corrosion, moisture, temperature effects, shrinkage and the use of incongruous construction materials and strategies [6–8]. Various structures including bridges, tunnels, railway tracks, roads, buildings, pavements, aircraft, and automobiles are prone to cracks [9]. Cracks are the earliest signs of degradation that can lead to serious damage if allowed to penetrate or left unmaintained or unrectified [9, 10].

Broadly, cracks can be described in terms of their occurrences time, width, component used for construction and overall activeness. Classical methods including SIFT, ORB, SURF etc. require extensive manual supervision and do not allow automated crack detection. Deep learning can reduce this overhead to identify the cracks, thus permitting the automatic labelling of whether the crack is active or dormant [11].

Once cracks have developed in a structure, they can either remain dormant or be active. The difference between the two types of progressions of cracks is that the dormant cracks stay unchanged throughout time period. The dormant cracks include a minor crack, thin crack, line-like crack, complex crack, and sealed crack. On the other hand, the active cracks progress with changes which include deepening of the width, increase in the length or spreading of the crack in more directions. Due to the direction and type of changes, the active cracks include reflection cracks, transverse cracks, and miscellaneous cracks [8]. Cracks in concrete structures can be described (and lead to) as the partial or complete segregation of concrete into separate parts upon fracture or breakage [12]. Therefore, an essential measure to sustain the structural safety and health of the engineered structures is the early detection of cracks through the utilisation of effective methods. The manual inspection of cracks is a tedious process that demands extra effort and time. It is also prone to subjective assessment of deterioration and inadequate observations by crack inspectors [13, 14].

The advancements and breakthroughs that have been achieved in computer vision and image processing techniques, enable the replacement of the manual crack detection methods with more effective automated inspection procedures [15]. The application of various computer vision-based techniques to efficiently deal with image segmentation [10, 16, 17], colour tracking [18], curvilinear structures [19] and crack detection [20–22], have been extensively reported in the literature.

Notably, the detection and localization of cracks are very complex, as numerous visual patterns are associated

with cracks and it is quite complicated to achieve a single method that can be applied to different cracks [10]. Using the crack detection methods or the traditional image processing techniques alone is not sufficient to deal with different scenes and for distinguishing cracks under different scenarios (i.e., lighting spots, shadows and edges). The literature indicates that deep learning based methods can be used for effectively overcome the limitations of traditional computer vision methods in terms of extraction and learning of high-quality features [12]. Therefore, the effective amalgamation of computer vision techniques with machine/deep learning approaches is highly necessitated to ensure the efficient and automatic detection and localization of cracks [23, 24].

The increased use of machine learning approaches such as neural networks instead of the traditional vision-based approaches has encouraged the exploration of other similar methods for crack detection in concrete. More recently, deep learning methods have been in focus for crack detection, particularly the Convolutional Neural Networks (CNNs) have gained considerable importance and applicability due to their high performance on many sophisticated computer vision tasks [25, 26] including image classification, image segmentation, and object detection [9, 10, 27, 28]. Traditional crack detection systems have a major limitation in that the applied method is highly specific to a particular situation or scene [11, 12]. In addition, various methods such as FoSA, FFA and CrackTree work considerably well for thinner cracks but fail when applied to wider cracks [13]. Moreover, detection of features fails at variable instances thus leading to a non-generalised extraction of features [13].

The use of CNN brings along various powerful hierarchical features, automatic feature learning, grid-like image topology, differentiation of multiple classes and improved detection of cracks without the requirement of additional image processing techniques [17]. The deep learning models also provide an improvement in the detection and classification performance by using the stacked convolutional layers for the exploitation of image features in different resolutions [29]. The pooling process and the presence of a set of sparsely connected neurons within the CNN require fewer computations as compared to ANN [29]. CNNs are designed to deal with visual data and capabilities including visual object recognition, object detection and image classification. CNN's one of the most efficient methods used for image recognition [12]. CNNs are more valuable than ANNs when it comes to visually processing information, with the latter being more inclined towards processing tabular and textual data. Also, CNNs are faster than ANNs when it comes to dealing with and sorting huge data sets.

Recent application of CNNs in literature includes the automatic detection of concrete cracks in roads, tunnels, and Gas turbines [13]. However, unlike other cases where the material surface is more homogeneous, the detection

of concrete surface defects (occurring on inhomogeneous surfaces) should be carefully configured in terms of deep architecture. This requires the use of an extensive data set and taking into consideration variable conditions leading to surface imperfection such as stress, cyclic loading, poor construction, deterioration/corrosion, moisture, temperature effects, shrinkage, and utilization of incongruous construction materials) that are essential for dealing with several real-world problems [22].

In this study, we propose a robust CNN-based classifier for detecting cracks in the concrete surface of bridges based on 10 convolution layers, and CycleGAN has been used to improve detection accuracy and avoid data augmentation. This method does not succumb to factors such as lighting, noise due to lighting, blur, casting, and shadow-based noise and provides wider adaptability. Unlike the traditional approaches, our proposed approach does not require the use of feature extraction and calculation rather it is capable of automatic learning of image features.

The paper is organized as follows. Section 2 presents the methodology, with brief overview of the case study, data collection and pre-processing of the images and the proposed methodology for the crack detection. In Sect. 3 the experimental analysis is elaborated, the performance metrics used for evaluating the techniques followed by the results and discussions. Section 4 summarises the key results of the study, performance of the suggested framework based on deep hierarchical CNN architecture along with Cycle GAN for predicting crack segmentation, and limitation of the study.

2 Materials and Method

In the current study, we propose the development of a robust CNN based architecture that includes a cycle generative adversarial network (Cycle-GAN) for detecting cracks on infrastructures such as bridges. Over the past years, CycleGAN has gained considerable progress in terms of utilisation in deep learning methods [30]. Therefore, due to the broader application of Cycle-GANs the detection of cracks on civil infrastructures can be dealt with effectively using image-to-image translation. Cycle-GAN provides network training without the requirement of ground truth labelling. Due to its capability of translating crack images to an image set that displays a pattern similar to the ground truth like images [18].

The proposed approach will assist in the robust, efficient, and cost-effective inspection of infrastructural health and the maintenance of infrastructural damages. Additionally, the proposed Deep Neural Network Framework for automating the crack detection process also provides an advantage of eased scaling to any edge device (i.e., coral dev, jetson nano). In this study, the proposed CNNs based architecture

was applied to the data set obtained from the Bolte Bridge in Melbourne, Australia.

3 Case Study

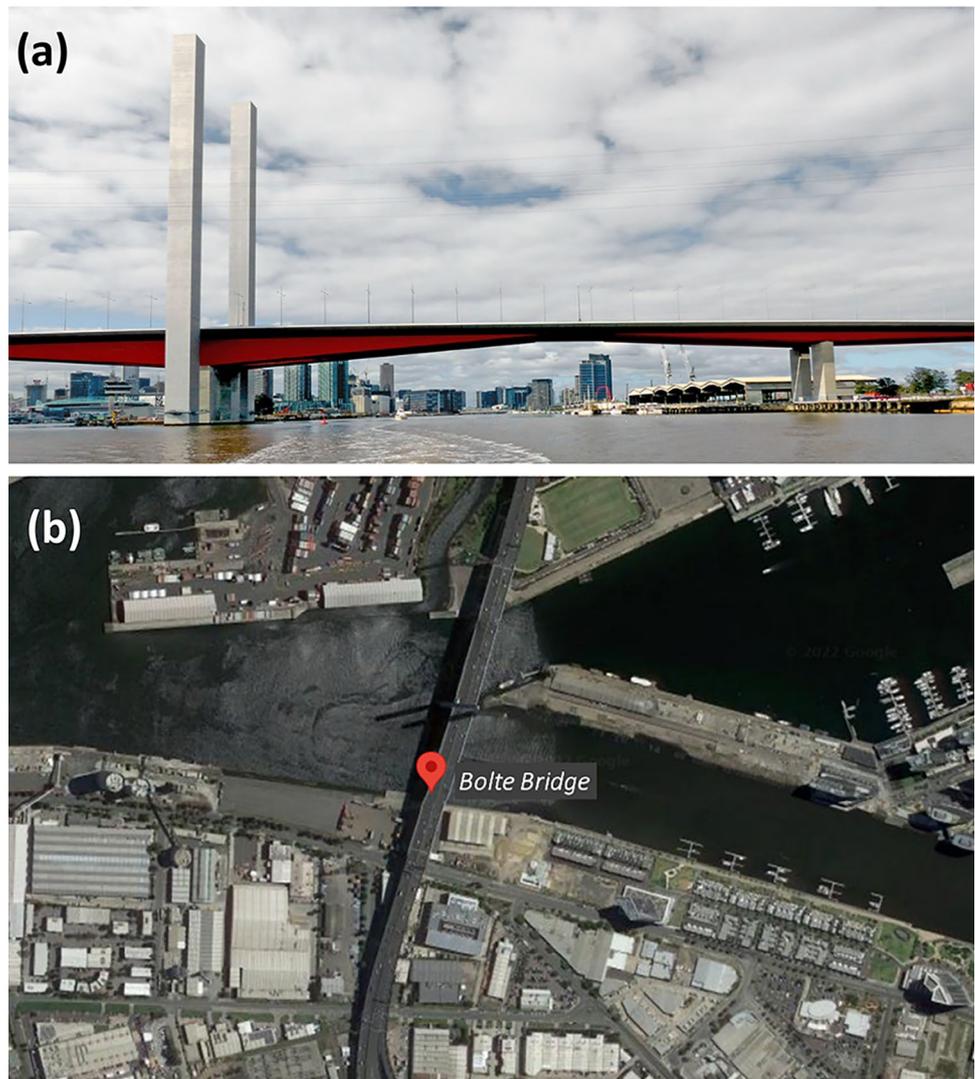
For the case study, the Bolte Bridge in Melbourne, Australia (Fig. 1a) was selected. Bolte Bridge is a large twin cantilever road bridge carrying a total of 8 lanes of traffic. It is present on the west side of central business district (CBD), spanning over the Yarra River and Victoria Harbour (Fig. 1b). The total length of the bridge is 490 m and comprises four spans, two sides of which are 72, long and the main measure 173 m. The data was collected by VERIS which is a leading company for providing spatial data services to their clients (Fig. 2a). VERIS provides an integrated approach for the project life cycle starting from the planning phase to the final delivery phase. It uses innovative technologies to conduct surveys and damage assessments of the infrastructures such as railways, bridges, roads, buildings etc. Aerial imagery of the Bolte Bridge was carried out using UAVs (unmanned aerial vehicles). A DJI M200 UAV was used for surveying the region (Fig. 2b). A machine learning-based algorithm was developed for crack detection. Images would typically be obtained from drones in cases where access is limited (e.g., due to the span of the bridge, presence of traffic or cases of floods), by automatically identifying cracks and vulnerabilities in the bridge infrastructure.

4 Data Collection and Pre-processing of Images

The crack detection procedure was initiated by the collection of 2D images that form the needed dataset. The model training and testing were performed on a single machine Intel Core i9-10900KF (10×3.70 GHz, 20 MB L3 cache, 125 W) with GPU (GeForce RTX 2080 Ti). The quantitative and qualitative results were observed and compared with state-of-the-art methods.

The images of the bridge were obtained using a UAV (Unmanned Aerial Vehicle) carrying a digital camera onboard (Fig. 2b). Besides that, images from public dataset CRACK9001 were gathered for training and testing purpose. A total of 2097 images were captured, with dimensions of 4864×3648. Images processed by deep learning are augmented through cropping, colour modification, geometric transformation, noise injection, and flipping. The images included in the dataset had three main types of cracks that can be classified into simple cracks, hairline cracks and artificial marking cracks as shown in Fig. 3. Simple cracks usually result from infrastructure settling onto its foundation however, in comparison, the hairline cracks are very small

Fig. 1 **a** The Bolte Bridge, Melbourne, Victoria. **b** Geographical location of the Bolte Bridge



and shallow that mainly emerge due to plastic shrinkages about 0.003 inches in width [31].

After finalizing the dataset, the collected crack images were preprocessed to remove any noise or undesirable background, following this step, an image brightness adjustment was carried out. Cropping was performed on the images to remove any unwanted background such as grass, water, sky, building, trees etc. Particularly, for the crack images, the data set was divided into two types of levels including the crack and structures without cracks (non-crack) levels respectively. The overall percentages of the pixels for all images (with or without crack) are shown in Table 1 which indicates that a lower percentage of the crack regions are included in the complete dataset.

A total of 2.93% significant crack pixels, 1.41% weak crack pixels and 95.93% non-crack pixels were included in the complete data set respectively. For both sets, training and test set, a total of 96.32% and 94.69% non-crack pixels

were included, as shown in Table 1. Additionally, a total of 3.24% and 4.15% of crack pixels for training and testing were used in the current study (Table 1). Generally, a crack width in the range of 1 to 5 pixels is considered a weak crack whereas significant cracks are those which have more than 5-pixel width. It was observed that the thin cracks and surface cracks had different properties in comparison to wider cracks. Therefore, the application of traditional post-processing methods (with length constraint, curvature and geometric features) is necessary to obtain the complete and continuous thin cracks [32], which is a limitation of the deep convolutional networks.

For the crack images in the current study, the height and width distributions are presented according to two levels mainly crack and non-crack respectively. Figure 4 illustrates the cracks in terms of spatial representation such that the width and height of crack pixels are gathered through Pytorch and WANDB [33]. Along with crack images, the

Fig. 2 **a** Field sampling day. **b** Specification of DJI M200

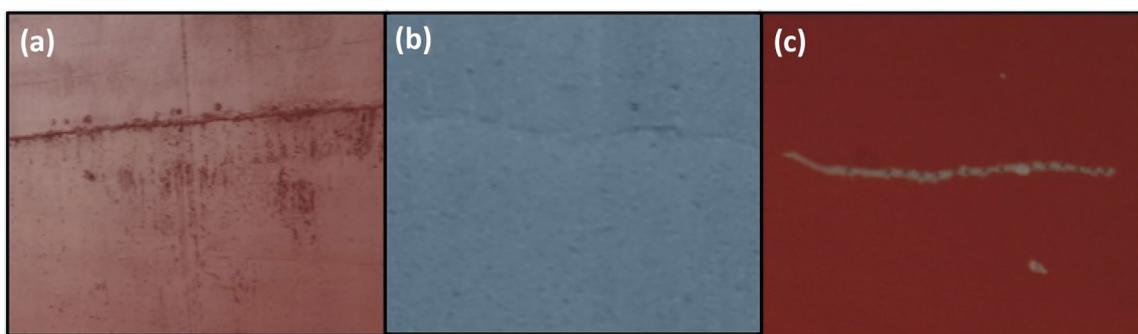
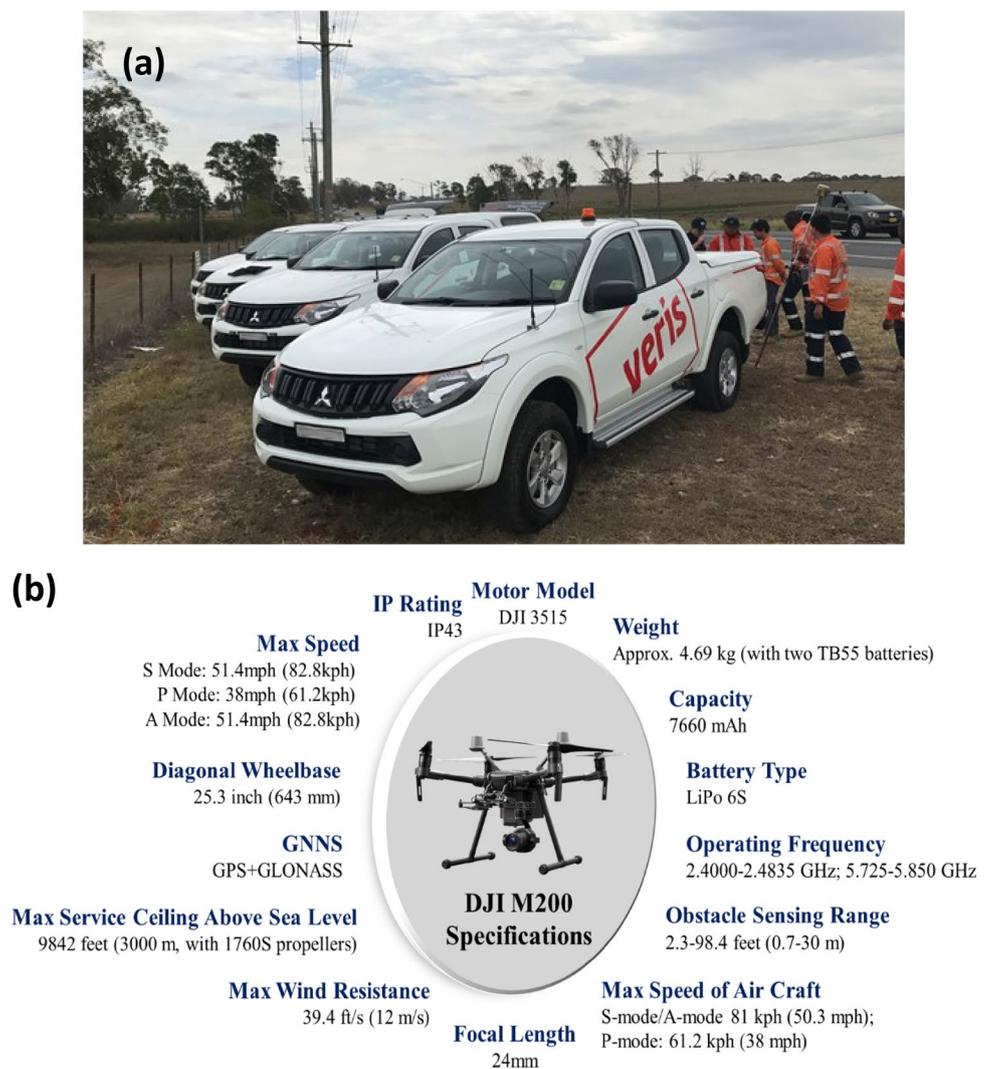


Fig. 3 The crack types used in the dataset include **a** simple crack, **b** hairline crack and **c** artificial marking crack

dataset also included pothole and water straining images for training and testing purpose.

For the current study, the crack pixels frequency was predicted, and the bounding boxes or labels were identified through spatial location analysis and the use of data

distribution. The axis presented in Fig. 5 provides the representation of size distribution and it is shown that spatial or frequency distribution for our crack pixels is neither skewed nor projected in one place. Rather, crack pixels display Gaussian or well-distributed pixel data as shown in Fig. 5

Table 1 Percentages of the pixels for crack and non-crack images

Pixels	Crack pixels (%)		Back-ground pixels (%)
	Significant	Weak	
Total	2.93	1.41	95.93
Training	3.24	0.44	96.32
Testing	4.15	1.16	94.69

which is indicative of the fact that the pixel distribution in the selected crack dataset is devoid of biases.

Many portions of the dataset consist of drive view images roughly 54% from road damage detection challenge 2020.

The total dataset for this study includes 10,000 cracks. The health of the dataset is explained through plots. The location is shown through a Gaussian distribution right

around the central region where most cracks appear. Determining the size of the crack is tricky because of the transformations that can occur in cracks. By looking at a crack, the only way to analyse the size of a crack with 100% surety is to be orthogonal to the crack. Moreover, the region of a crack that is close to the camera is fully visible but the ones further away from the surface may appear like a thin edge or depict other differences due to the camera angles as well as the transformations in the crack which might make this difficult to detect. The proposed methodology can potentially enable crack analysis in terms of structure in a consistent manner. This method can generalize to the environmental setting but cannot gauge shifts in perspective. The comparison of generative and discriminative methods is shown in Table 2.

Figure 5 shows a comparison of generative and Discriminative results of Faster-RCNN and Yolov5-s). The highly expressive Deep CNNs entailing numerous parameters

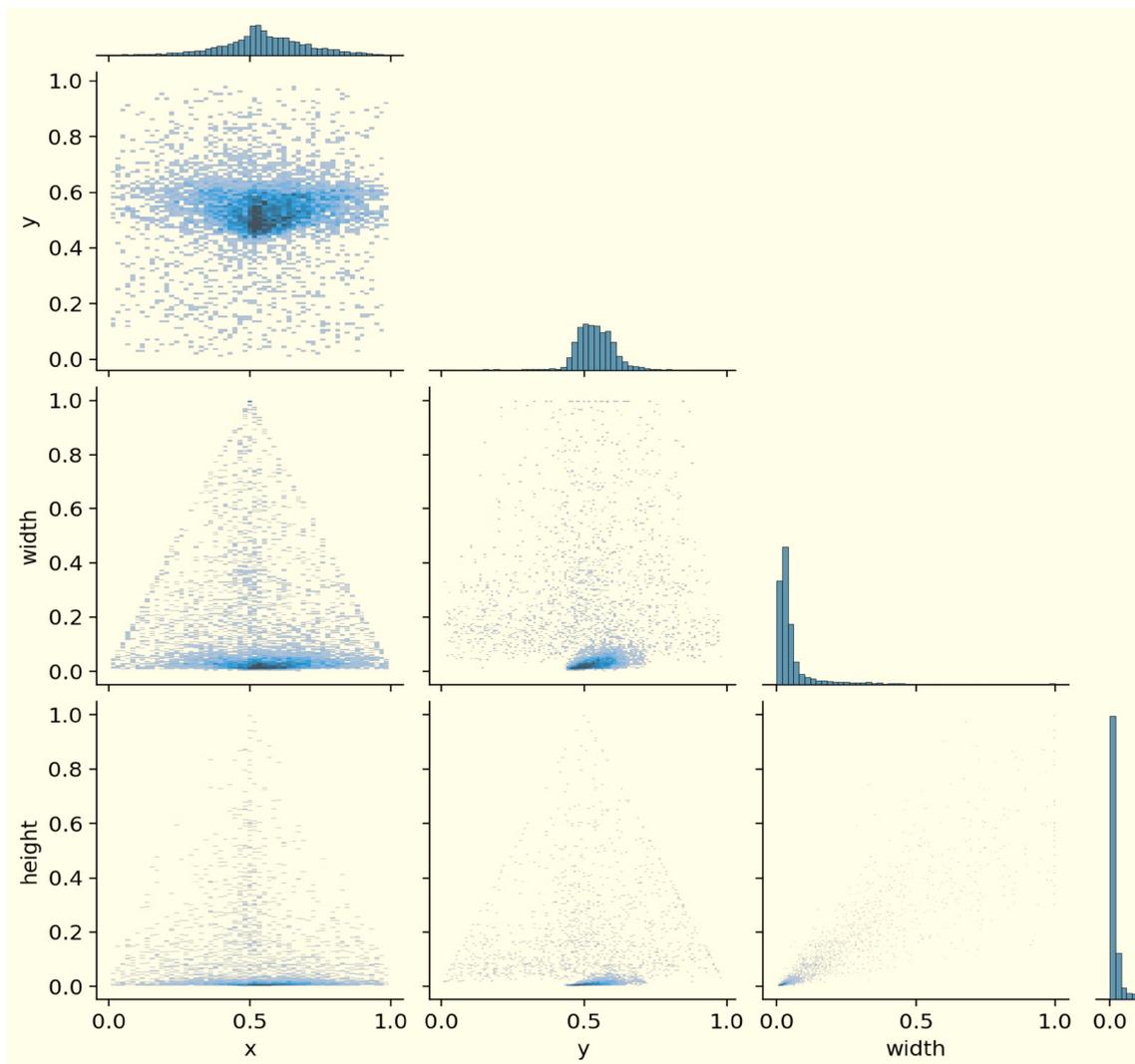
**Fig. 4** The height, width, and spatial extent of the crack pixels in our dataset

Fig. 5 Comparison of generative and discriminative methods

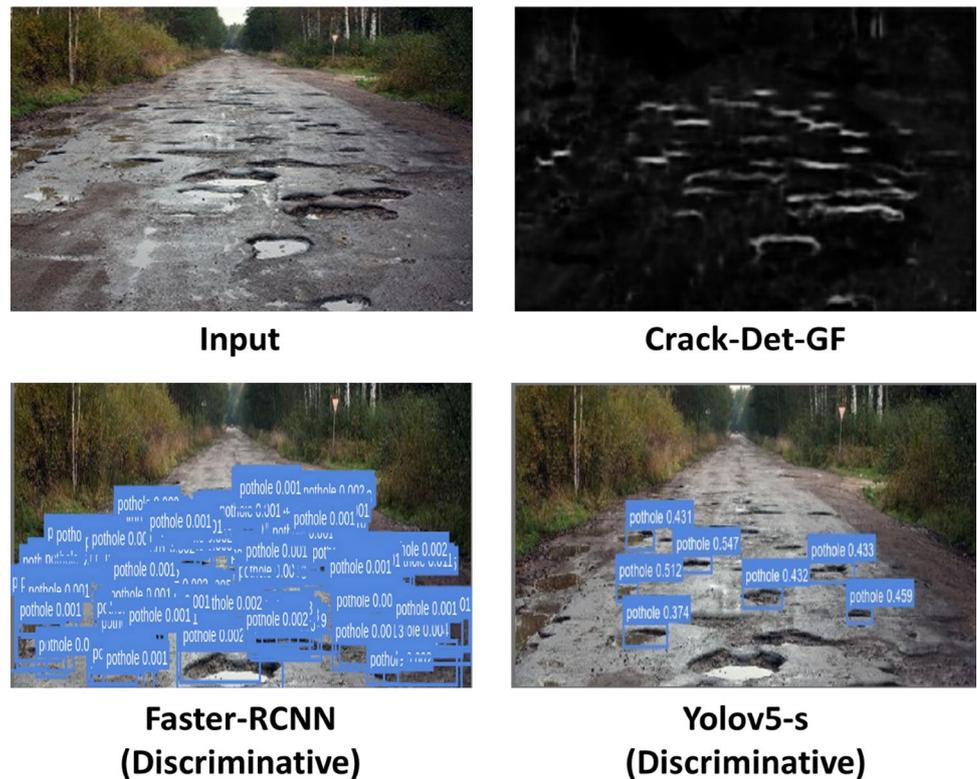


Table 2 Comparing generative and discriminative methods

Characteristics and approach	Generative	Discriminative
Learns	Latent Code	Mapping ($X \rightarrow Y$)
Inference speed	Slow	Fast
Generalization	Moderate	Poor

have brought considerable advancements in the classification and processing of images [29]. However, the image features in the CNN’s training set can be a risk as it tends to over-fitting because of the non-generalized features in this network. Using an insufficient set of samples for training can lead to overfitting [29]. Additionally, the collection of abundant samples is an exorbitantly costly endeavour, which has increased the utility of data augmentation methods (i.e., flipping, resizing, random cropping) to enhance image variation and overcome the issue of over-fitting [34]. In the overall training procedure of the proposed approach, label generation and crack detection were performed through data augmentation are presented in the Table 3.

5 Proposed Method

The overall workflow of the current research study is presented in Fig. 6.

TABLE 3 Data augmentations details

Augmentations	
HSV—hue augmentation (fraction)	HSV_H: 0.015
HSV—saturation augmentation (fraction)	HSV_S: 0.7
HSV—value augmentation (fraction)	HSV_V: 0.4
Rotation (\pm deg)	Degrees: 0.0
Translation (\pm fraction)	Translate: 0.1
Scale (\pm gain)	Scale: 0.5
Shear (\pm deg)	Shear: 0.1
Perspective (\pm fraction), range 0–0.001	Perspective: 0.2
Flip up–down (probability)	FLIPUD: 0.0
Flip left–right (probability)	FLIPLR: 0.5
Mosaic (probability)	Mosaic: 1.0
Mix-up (probability)	Mixup: 0.0

5.1 Per-pixel Segmentation

The use of the pre-trained model for semantic segmentation does not work on general images because it is based on the association of a class label to each pixel of an image. Therefore, we used Crack9001 (A publicly available crack-detection dataset) for the training of the SegNet which aims to perform pixel-wise segmentation of the captured dataset (by UAV). The SegNet method displays limited accuracy and requires manual supervision therefore, per-pixel annotation was used in the current study.

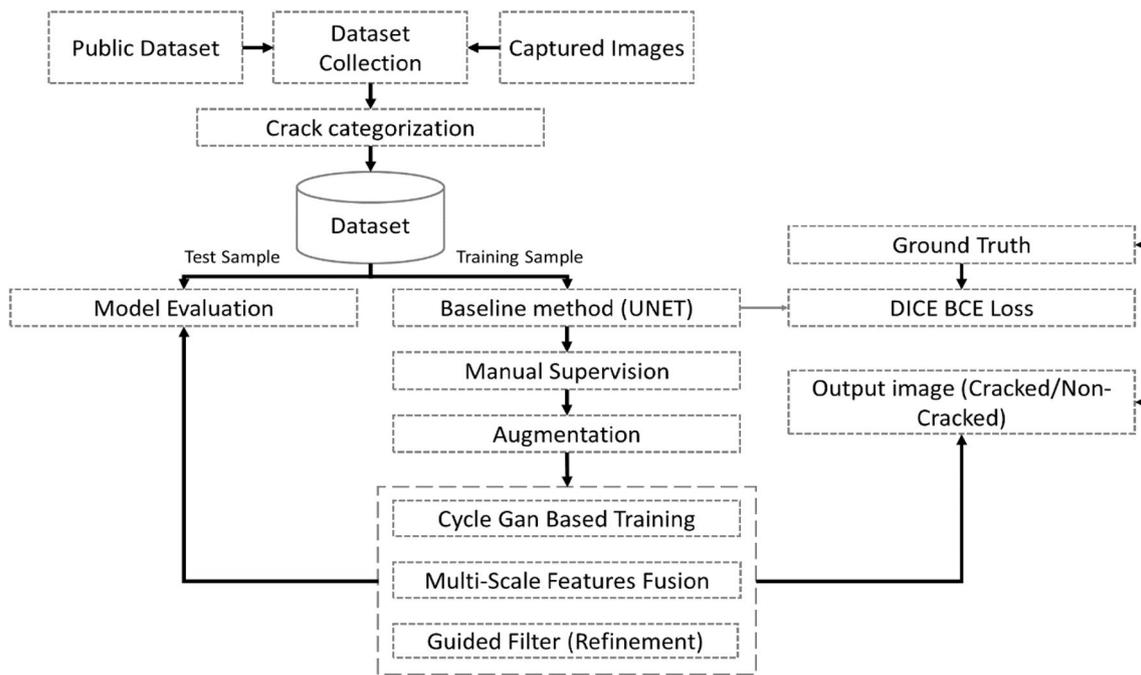


Fig. 6 A holistic view of the framework proposed in the study

5.2 Baseline Design (BN)

It consists of Max Pooling, ReLU Activation, Concatenation, and convolution operation (Fig. 7). It consists of three sections i.e., contraction, bottleneck, and expansion. For obtaining high precision results in semantic segmentation, it is vital to collect finer details while retaining semantic information. However, having a limited dataset for training a deep neural network is a limitation. This can be overcome by using a pre-trained network and applying it to the desired datasets. The extensive data augmentation carried out in U-Net is another way to overcome the rainy issues. Its key contribution is the creation of shortcut connections. The performance of the U-Net can be enhanced by replacing the plain unit with the residual unit.

5.3 Training

5.3.1 Loss Function

Boundary loss for road boundaries (highly unbalanced segmentation) is being used. The loss function aims to get smoother outputs at the boundaries and enhance model output for two close parallel roads. The integrals are used over the boundary between the regions by boundary loss, instead of applying the unbalanced integrals over the regions. The boundary loss function was used in combination with BCE-Dice Loss. Learning Rate, Epoch loss and Epoch IoU Score Plots are shown in Fig. 8.

The model was initially trained for the first 10 epochs with a combination of boundary loss and BCE-dice loss (Fig. 8) and further fine-tuned for another 30 epochs (Fig. 8). The cyclic learning rate was used with a cycle size of 5 epochs and a learning rate decay of 0.8 (20%) after each cycle. To ensure that only the best weights were used during inference, an early stopping criterion was applied. It was observed that when network was evaluated on unseen datasets a loss in performance was observed. While better performance was achieved when evaluated for synthetically modified dataset.

The existing methods for crack detection face many limitations, which mainly include the availability of limited datasets. Changing the dataset in such cases leads to difficulties in crack detection. Examples include CrackTree, FFA, and FoSA methods which are reliable for thin cracks but tend to fail in terms of detecting wider cracks. These applications stand to benefit from the hierarchical features and powerful abilities of CNN. The use of CNN is suitable for the goal of learning a non-linear model for image analysis. A Conditional Random Field (CRF) has been used previously [16] for refining low-resolution images as a post-processing step. On the other hand, using a Fully Convolutional Network (FCN) results in up sampling of the feature maps but the output of such a method is not very accurate. Hence, using an approach that combines the parameters included in CNN, CRF and FCN was considered more desirable.

In this architecture, there are no fully connected layers, the side-output layers are inserted after the convolutional

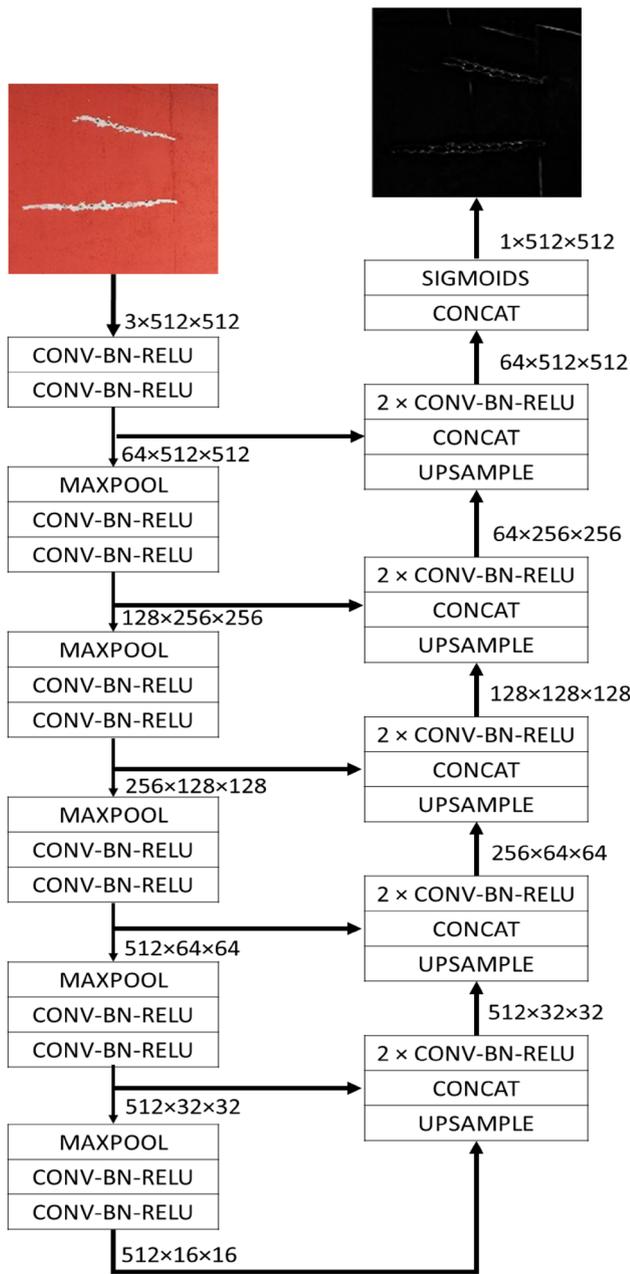


Fig. 7 Proposed deep residual UNET architecture

layers, deep supervision is applied at each side-output layer and then all of them are concatenated to form a final fused output (Figs. 9, 10 and 11). In this way, the final output layer acquires multi-scale and multi-level features as the plane size of the input of side-output layers becomes smaller and the receptive field size becomes larger. The fused prediction is refined by guided filtering with the first side-output layer (Fig. 11).

Predictions made at the processing stages can preserve the boundaries of cracks but are also sensitive to noises such as dark spots and dirt. On the other hand, better

anti-noise capabilities are shown by the predictions of deeper convolutional stages. However, a failure in the preservation of segmentation boundaries is also associated with predictions of deeper convolutional stages. Therefore, it is commendable to carry out a linear combination of all the combinations carried out at different stages. We added some modules for refinements such as phase shift and convolutional layers [10]. The binary mask is generated initially which is followed by the setting of the guidance map with a side-output (conv1_2). The final refined prediction significantly preserves the crack boundaries also leading to noise removal in the low-level prediction. Additionally, for training, we have also used the cycle-consistent generative adversarial networks (Cycle-GAN) that can reduce human intervention for manual label generation. The guided filter achieves the final refined prediction by well preserving the crack regions and removing the noises in the low-level prediction. Compared to the CRF method, such a technique is faster and more efficient.

5.4 Model Training Using Cycle-GAN

In recent years, Generative adversarial networks (GAN) have been employed in deep learning methods successfully as it offers a novel strategy for the training of different models [34]. Originally for GAN, a fully connected layered generator configuration is used that allows the images to be generated from random noises. However, lately, the cycle-consistent adversarial networks (Cycle-GAN) were proposed by Zhu, Park [35] which allowed effective training without the need for the using data pairing step. Therefore, based on the applicability of Cycle-GANs we develop crack detection in concrete structures as a translation problem in the image-to-image translation approach. Notably, the Cycle-GANs can effectively train the networks without the requirement of manually labelled Ground Truths, as they enable images with similar outlooks to be translated [36].

For the Cycle-GAN based training of the network, two separate data sets are required (Fig. 12). These include the crack image set (M) with images {mi}, and the structure library (K) with {ki} images respectively. The network topology is based on two image-to-image translation GANs (i.e., Forward and Reverse GANs) as presented in Fig. 12. The forward and reverse GANs perform image translations from $MtoK(F : M \rightarrow K)$ and $KtoM(F : K \rightarrow M)$ respectively. The system contains two discriminators mainly Dm and Dk . Here, Dm is used for distinguishing between the {mi} and {R(Ki)} with L_{advr} (reverse adversarial loss) whereas, to overcome the data imbalance and differences in domains, Dk is used that distinguishes between {ki} and the translated images{F(mi)}. The objective function is presented in Eq. 1:

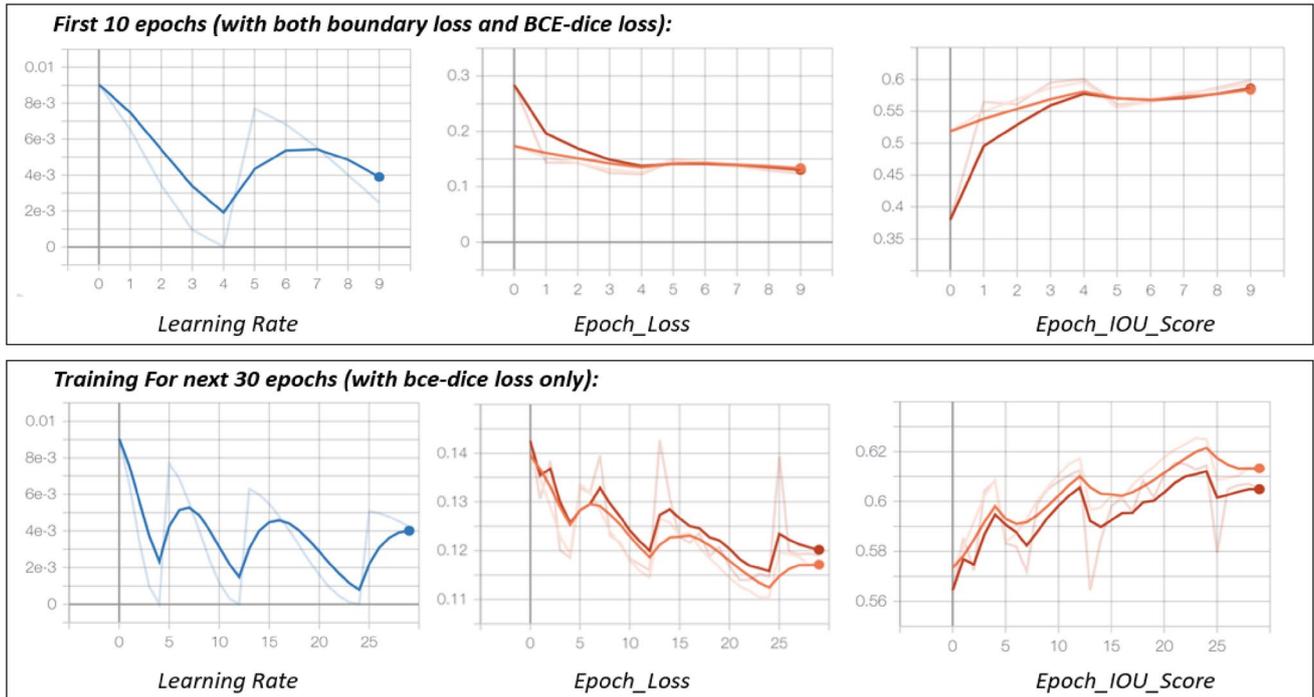


Fig. 8 Learning rate, epoch loss and epoch IoU score plots (red and orange line implies validation and training data)

Fig. 9 The multi-layer CNN architecture used for damage detection

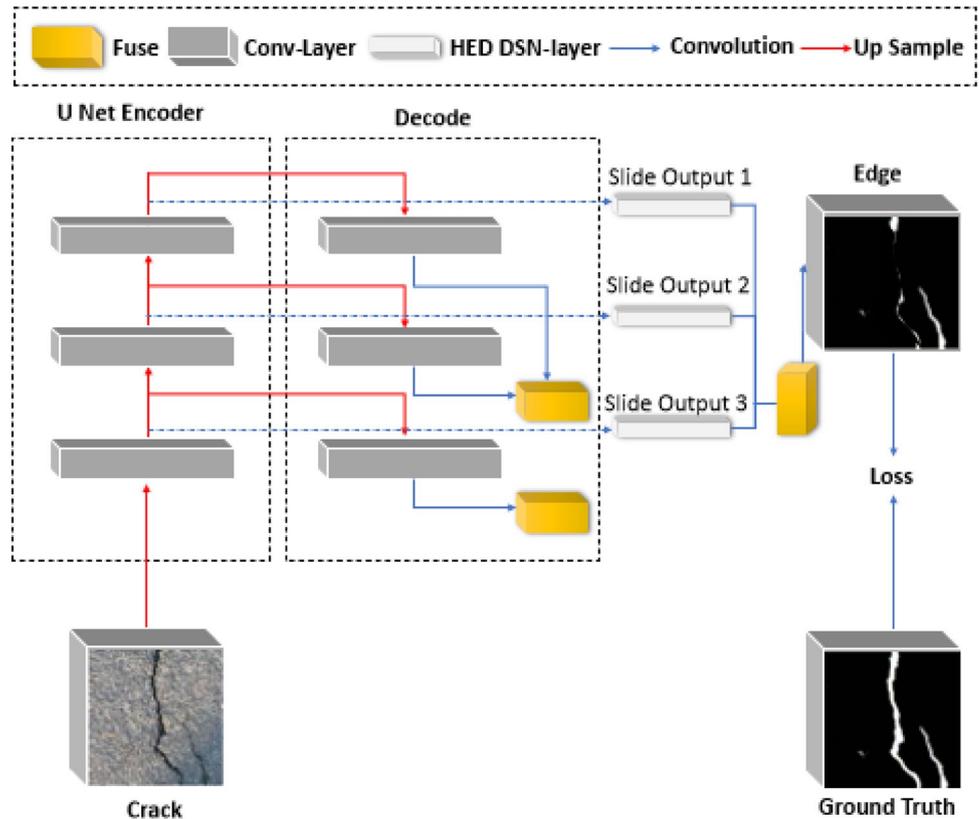


Fig. 10 Encode-decoder architecture used for damage detection

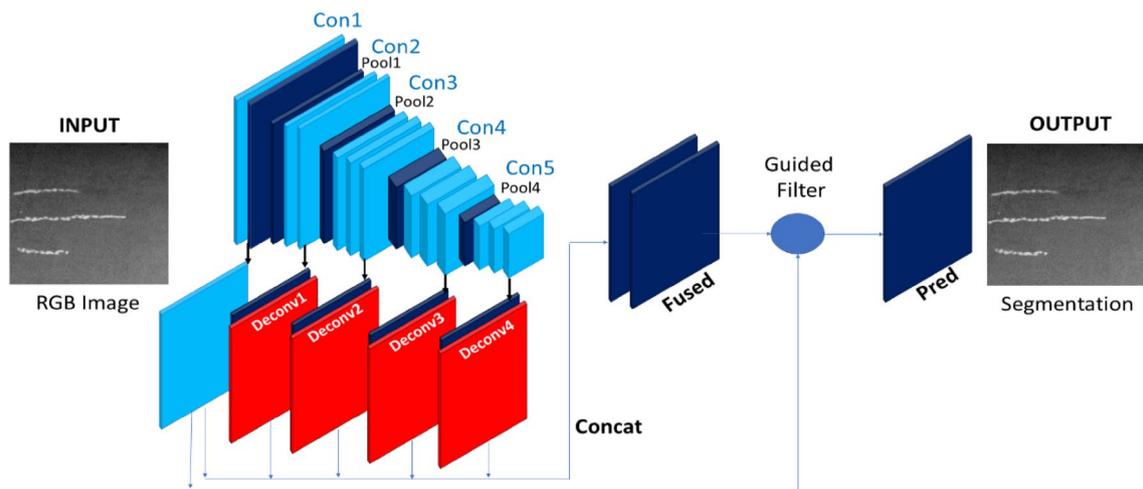
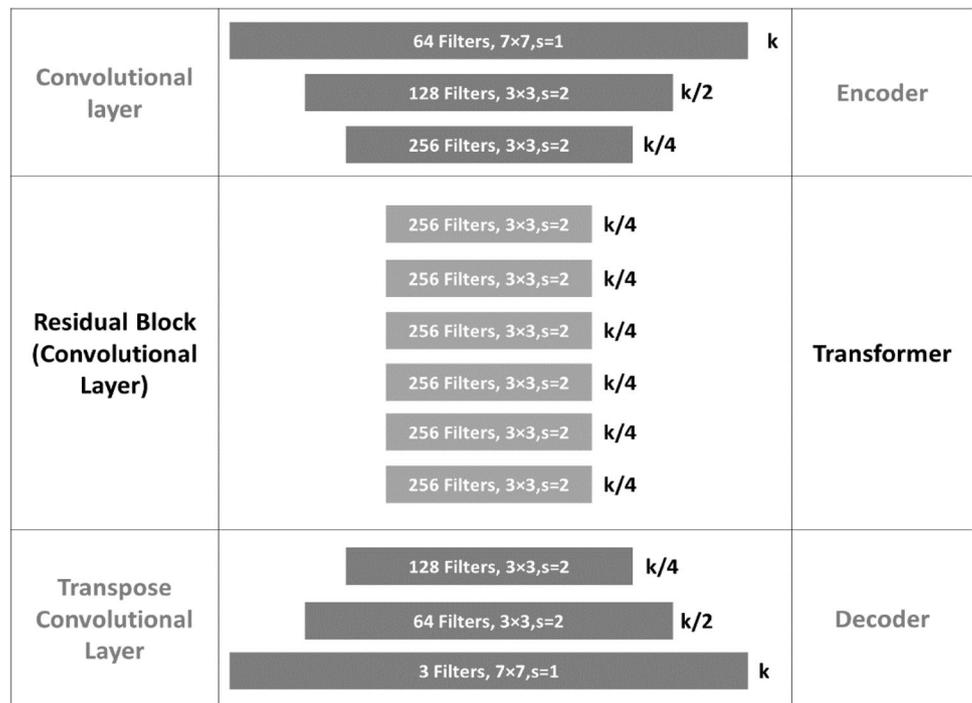


Fig. 11 Deep crack architecture

$$L = (L_{advf} + L_{advr}) + \lambda(L1_{fc} + L1_{rc}) \tag{1}$$

Here, λ controls the weight between the two losses (adversarial and the cycle-consistent loss), and $L1_{fc}$ and $L1_{rc}$ represent the two-cycle consistent losses with L1-distance formulas in the forward and reverse GAN respectively [30].

5.4.1 Adversarial Loss

Real-like images can be generated from noise while using generative adversarial networks for training. The GANs

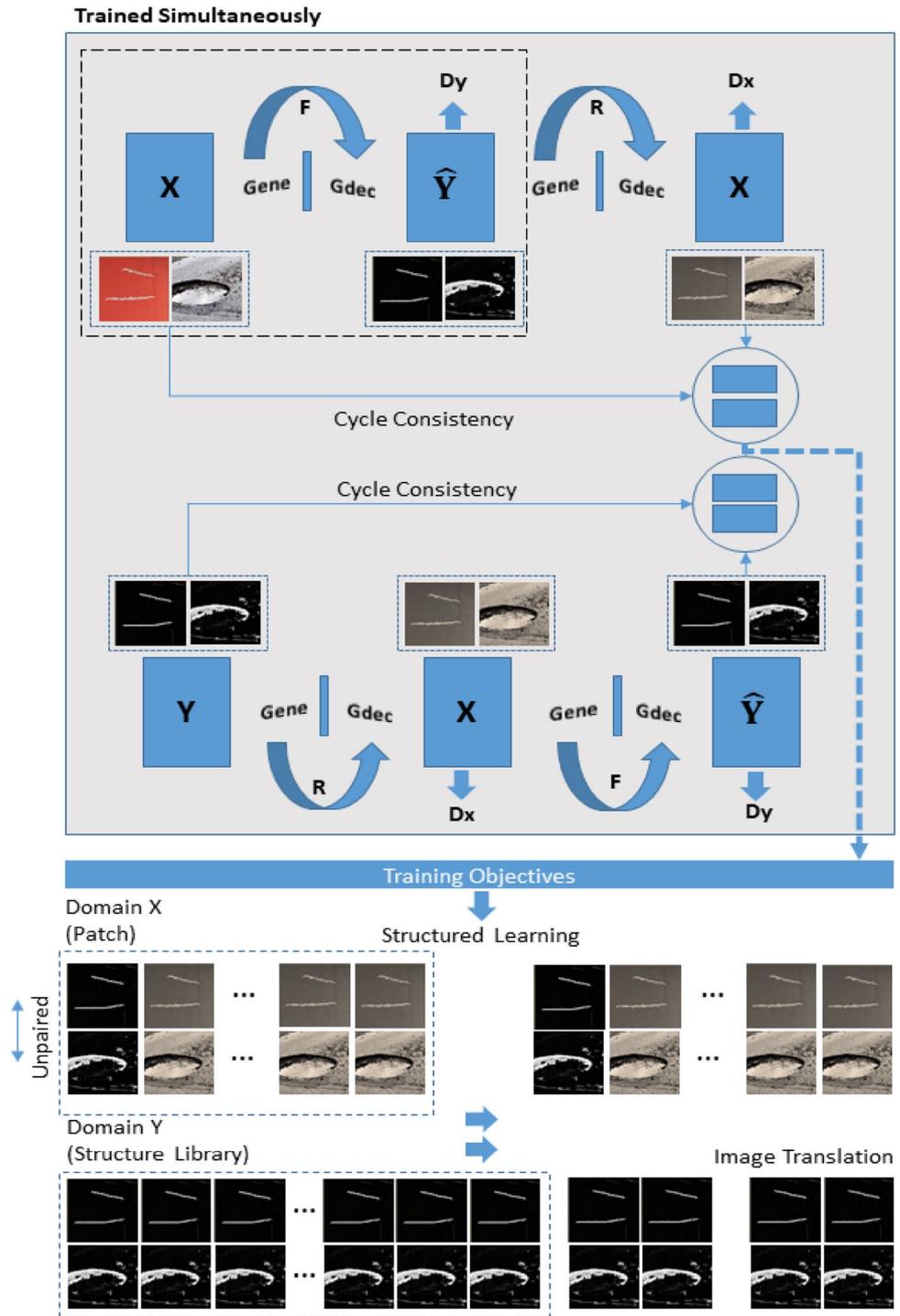
execute by max–min two-player game and it is, therefore, important to alternately optimize the following objectives (Eqs. 2 and 3):

$$\max_D V_D(D, G) = E_{yP4^{(y)}} [\log D(y)] + E_{xP4^{(x)}} \tag{2}$$

$$\max_G V_G(D, G) = E_{xP4^{(x)}} \tag{3}$$

Here, D, G, x and y denote the discriminator, generator, noise vector input into the generator and real image in the training set respectively. G Generates images (Gx) that are

Fig. 12 Use of cycle-Gan for Training



like images from Y and D distinguishes between the real samples 'y' and the generated sample $G(x)$. Moreover, D and G try to maximize Eqs. 2 and 3 respectively, which results in adversarial learning.

5.4.2 Cycle-Consistency Loss

It is well known that the adversarial loss can help in obtaining structured images, but when used alone it is inadequate

for translating the crack image patch to the desired structure patch or the other way round. Thus, it does not guarantee the consistency of the structure pattern between the input and the output images. Therefore, the introduction of an extra cycle consistency parameter can help in training the CNN and maintaining the consistency of structure patterns between the input and output [31]. For the data set Q , each sample ‘ q ’ should be able to return to the original patch through the network, after the processing cycle ($q \rightarrow G(q) \rightarrow F(G(q)) \sim q$). Similarly, for each structure image, ‘ s ’ in the structure set the network should allow the return of n back to the original image ($s \rightarrow R(s) \rightarrow F(R(s)) \sim s$). These constraints can lead to the formulation of cycle-consistency loss defined as follows (Eq. 4):

$$L_{cyc}(F, R) = E_{Qp^4(q)} \quad (4)$$

5.5 Model Parameters

The CNN was developed on sophisticated implementations including FCN [26], DSN, HED and SegNet whereas, the CRACK9001 library was used for training [13]. Stochastic Gradient Descent (SGD) was used for optimizations. The aim here was to differentiate between two classes (crack, and non-crack) and utilize the loss of function, normalization, and side-output layers for the network such that they can provide enhanced accuracy and convergence along with eliminating the need to use networks based on pre-trained models. The model parameters selected for the study were (i) the size of the input image was $544 \times 384 \times 3$ (ii) ground truth size $544 \times 384 \times 1$ (iii) learning Rate 1×10^{-4} (iv) loss weight associated with each side-output layer was 1.0 (v) loss weight associated with final fused layer was 1.0 (vi) momentum 0.9 and (vii) weight decay was 2×10^{-4} .

5.6 Data Augmentation

Data augmentation forms an integral component of deep networks. The data set was augmented 10 times for this study. The data augmentation was carried out by (1) rotating images to 12 different angles after each 30° in. $[0^\circ, 360^\circ]$, (2) editing the largest rectangle without blank regions in the rotated image, and (3) horizontal flipping of images at each angle. However, for training the network both raw and augmented images were used and due to rotation transformations, resized input images (256×256) were used.

6 Experimental Analysis

The database was analysed using the selected methods and performance was evaluated based on the metrics and F-score. The results of the proposed architecture were compared with existing methods for the crack detection.

7 Performance Metrics

The proposed architecture was applied to the collected database and three metrics were used for the evaluation of common semantic segmentation [22]. We calculated the Global accuracy (GC), class average accuracy and the mean intersection of the union over all classes. The global accuracy estimates the percentage of correctly predicted pixels and is calculated in Eq. (5) as follows:

$$GC = \sum_m n_{mm} \sum_m t_m \quad (5)$$

The Class average accuracy (CAC) measures the predictive accuracy over all the classes and is defined as follows (Eq. 6):

$$CAC = \left(\frac{1}{n_{cls}}\right) \sum_m n_{mm}/t_m \quad (6)$$

Whereas the mean intersection of the union (IoU) over all classes is calculated using Eq. 7. IoU metric is used for the quantification of percent overlap evident between a target mask and the predictions made for output results. Briefly, the IoU parameter can be used for the measurement or quantification of the number of overlapping pixels between a target mask and the predictions made for output results [12, 32].

$$IoU = \left(\frac{1}{n_{cls}}\right) \sum_m n_{mm}/(t_m + \sum_p n_{pm} - n_{mm}) \quad (7)$$

7.1 F1-Score

In addition to the measures, three other metrics including Precision (P, Eq. 8), Recall (R, Eq. 9) and F-score (F, Eq. 10) were also calculated to evaluate the semantic segmentation. The Precision (P) parameter indicates the positive predictions for a positive class whereas the R metric is utilized for the quantification of positive predictions for all the positive classes included in the collected dataset [12, 32]. Moreover, the F-score is a measure that considers the precision and recall parameters. The F1-score metric indicates a model’s accuracy on a considered data set. Figure 13a, b represents the confusion matrix and the obtained results.

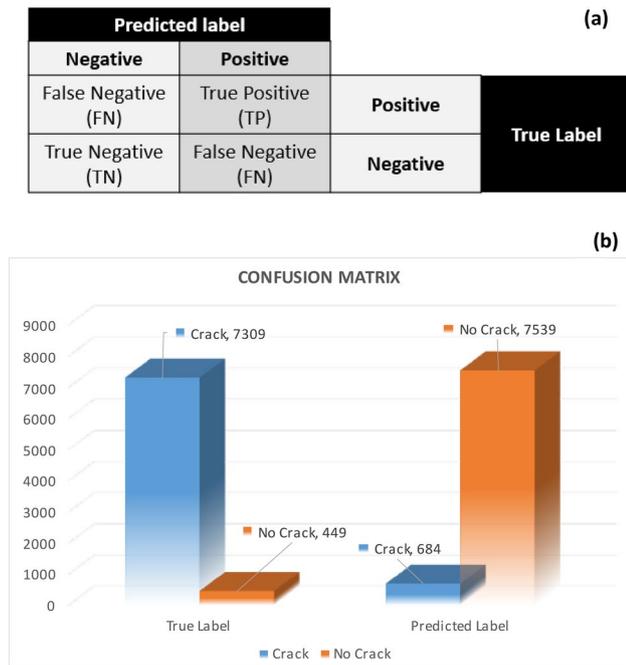


Fig. 13 **a** The confusion matrix. **b** crack and no crack images as per matrix

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \quad (8)$$

$$TPR = \frac{TP}{TP + FN} \quad (9)$$

$$TNR = \frac{TN}{FP + TN} \quad (10)$$

$$PPV = \frac{TP}{TP + FP} \quad (11)$$

$$NPV = \frac{TN}{TN + FN} \quad (12)$$

$$F1 = \frac{2TP}{2TP + FN + FP} \quad (13)$$

8 Results

8.1 Performance Analysis of Proposed Methods

We compared our method to three other common methods adopted to corroborate our experiments. The methods considered were the (1) Crack-BN, (2) Crack-GF and (3) SegNet [23]. Additionally, SegNet is also one of the latest

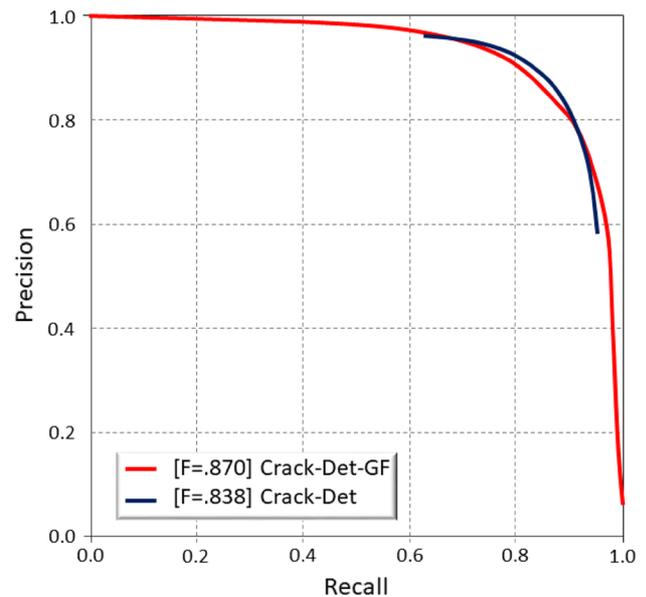


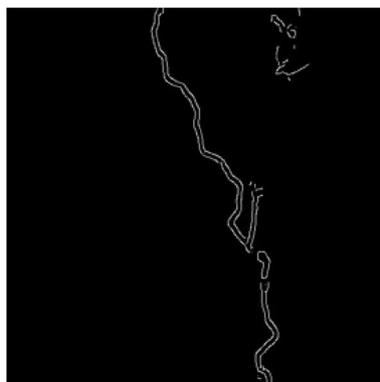
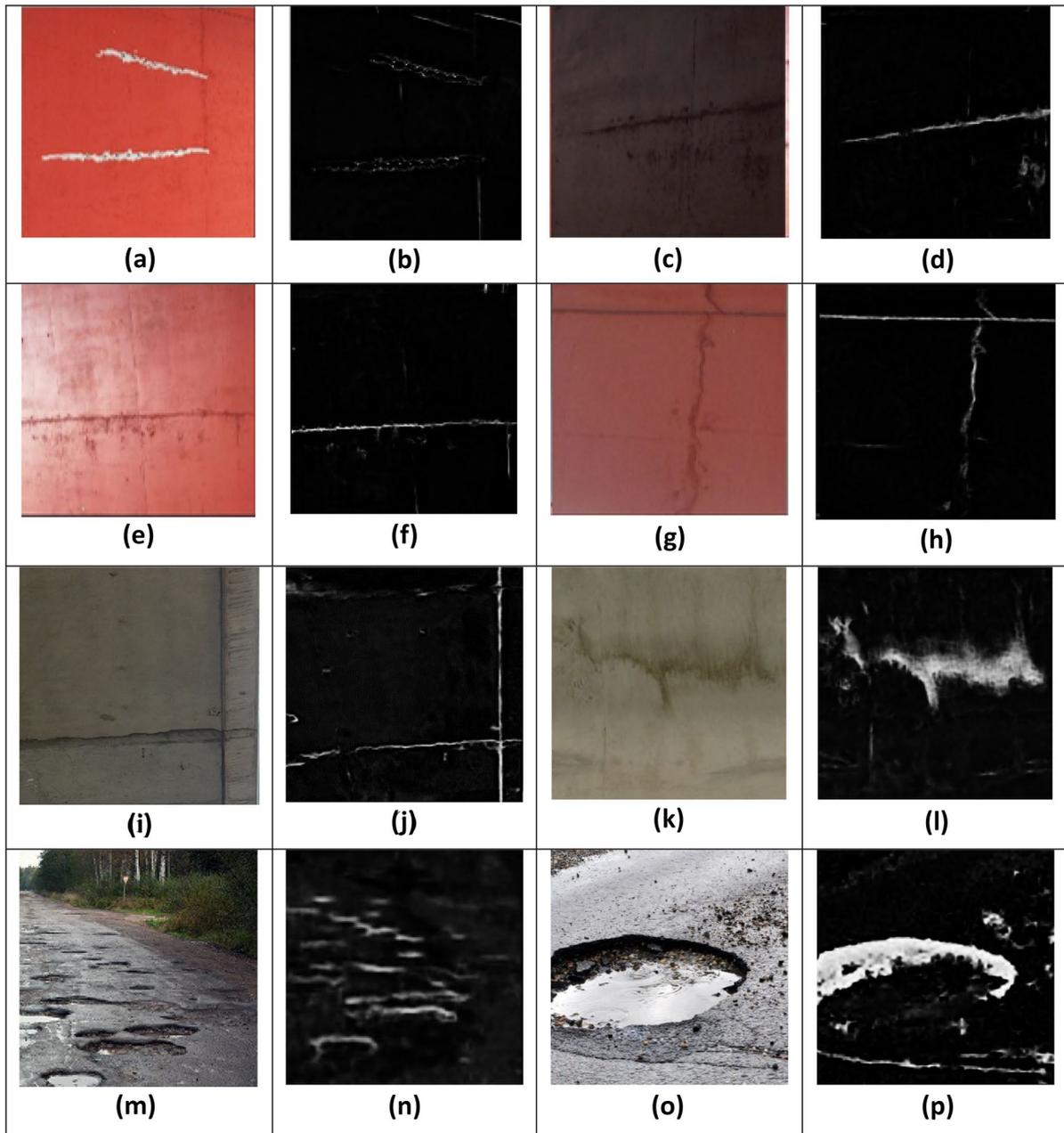
Fig. 14 Precision and recall Crack-Det and Crack-Det-GF

approaches that are used to perform semantic segmentation. Here, fine-tuning of the SegNet network and loss functions was carried out on the augmented datasets that were used in the current study. Crack-BN is also based on HED [24]; before operation activation, additional batch normalization layers are added. In Crack-GF, a guided filtering method that is highly efficient and rapid as compared to the conditional random fields (CRF) is used [37]. The probability maps were also binarized using the variant global thresholds. The Precision-Recall curves created through segmentation methods used in the present study are presented in Fig. 14. A representative data set of images obtained from target structures and the segmentations utilized are presented in Fig. 15.

The P–R curves generated for the two methods show that the GF method has a better performance as can be seen by the F-Score value (0.870) in contrast with the baseline method exhibiting an F-Score of 0.838 as stated in Fig. 14.

Our proposed architecture shows a significant upgrade in the performance, in comparison to the existing methods (Crack-BN, Crack-GF, and SegNet) included in the study as depicted in Table 4.

Batch normalization can significantly boost the performance as it leads to reduced over-fitting for the CNN. Moreover, the dense predictions can be refined through the implementation of both the guided image filtering and conditional random field methods. Results show that guided image filtering appears to be faster and more efficient in comparison to other techniques. It is notable from our results that in comparison to all other methods our proposed GF pipeline displayed the highest-Class Average Accuracy, Mean IOU, F-force, Global Accuracy, Recall and Precision values of



1:15	2:15	3:14	4:3	5:25	6:12	7:15	8:7	9:12	10:15
11:14	12:13	13:15	14:1	15:1	16:12	17:13	18:2	19:15	20:13
21:15	22:15	23:12	24:15	25:0	26:15	27:15	28:2	29:12	30:15
31:15	32:12	33:12	34:15	35:0	36:15	37:12	38:3	39:14	40:13
41:12	42:15	43:10	44:10	45:0	46:0	47:14	48:12	49:15	50:12
51:10	52:11	53:12	54:10	55:10	56:0	57:12	58:15	59:14	60:10
61:11	62:10	63:15	64:12	65:11	66:0	67:14	68:12	69:13	70:13
71:3	72:13	73:13	74:15	75:12	76:0	77:1	78:13	79:13	80:10
81:0	82:10	83:12	84:13	85:14	86:1	87:0	88:15	89:14	90:10
91:10	92:15	93:12	94:12	95:15	96:0	97:1	98:10	99:15	100:15
101:10	102:15	103:12	104:10	105:15	106:15	107:0	108:15	109:15	110:10
111:10	112:10	113:12	114:10	115:15	116:13	117:0	118:12	119:14	120:10
121:12	122:10	123:10	124:15	125:10	126:10	127:0	128:15	129:12	130:10
131:15	132:12	133:10	134:15	135:10	136:13	137:0	138:1	139:10	140:12
141:12	142:15	143:12	144:15	145:15	146:15	147:0	148:0	149:13	150:10
151:13	152:15	153:12	154:15	155:12	156:12	157:0	158:12	159:10	160:15

Fig. 15 Cracks categorization matrix distinguish crack pixels based on the pixel width

Table 4 Comparison between the methods

Outputs	Global accuracy	Class average accuracy	Mean IoU	Precision	Recall	F-score
Crack-Det-GF	0.989	0.931	0.878	0.881	0.861	0.870
Crack-Det	0.983	0.899	0.902	0.853	0.825	0.838
Crack-BN	0.980	0.876	0.872	0.845	0.811	0.827
Crack-GF	0.978	0.871	0.862	0.837	0.806	0.810
SegNet	0.967	0.867	0.822	0.768	0.760	0.786

0.931, 0.878, 0.870, 0.989, 0.861, and 0.881 respectively (Table 4). However, in comparison to all other methods, the lowest performance was achieved using SegNet as indicated by the statistical parameter values presented in Table 4.

Most importantly the total crack pixels used for e training and testing were divided into significant and weak crack pixels respectively. This categorization was used for distinguishing crack pixels based on the pixel width. A crack having a score between 1 and 5 for pixel depth was defined as a weak crack whereas; a crack exhibiting a pixel width greater than 5 was defined as a significant crack pixel as shown in Fig. 15 [21].

About metric values, our method generalizes better than the respective Crack-BN, Crack-GF and Seg-Net as shown in Table 4. Effective data augmentation methods are essential for deep models when training data is very limited. Moreover, using refinement modules like the PS operation and convolutional layers for analysing the overlap between the two maps shows that the proposed method can provide higher generalization and retain a greater amount of information on the low-dimensional features (Fig. 12). The results of this study show that this model can perform more robustly as compared to other methods. Moreover, our method also removes the background and irrelevant noise in the dataset [12].

9 Discussion

In this study, a drone was used for capturing images of cracks on concrete bridge surfaces (Fig. 2b). A total of 2097 images were captured. The total data set of images were divided into two, the training and test sets containing 1300 and 237 images respectively. Overall, 78% of the images containing a significant crack and 13% of images with a weak crack were used. However, 9% of the non-crack images were used in the test set only.

We have presented some of the corresponding segmentation and representative images used in the study in Fig. 15. The segmentations were generated by representing the subject in the binary images. For every image, a pixel-wise segmentation map was utilized which allows coverage of the crack regions and the pixel size of images were readjusted

to 544×384 . The percentages of the pixels for crack and the non-crack images are shown in Table 2. For each image, a pixel-wise segmentation map was used, which represents the total crack region coverage in the collected image set. To include a universal representation of cracks in the current study, a diverse range of scenes and scales were considered to select the crack images.

In our proposed framework loss of function, side output, and batch normalization were used to distinguish the crack and non-crack levels in the current study. However, to reduce the overfitting of the proposed CNN, a performance boost can be achieved through batch normalization. Model parameters including Loss weight of each side-output and final fused layer, momentum, decay, and learning rate were used for the training of our CNN network. Additionally, to reduce training time, a small dataset was used to train our CNN model. Most prominently, our network was trained using two different approaches including (1) baseline (BN), and (2) Guided Filtering (GF). In the baseline approach, data augmentation is not performed. However, the baseline pipeline is based on the UNET and our modified loss functions for smooth training. In later design, we have also added batch normalization layers before each activation operation to address domain invariances and co-variance shifts. The GF is a version of the baseline with the application of a guided filtering module. For every approach, we have used an augmented dataset for training.

Overall, in the current study, we assessed the performance of the proposed baseline and GF methods. Additionally, the performance was also compared with three other methods namely, Crack-GF, Crack-BN and SegNet. The performance of the studied methods was measured using Mean IOU, F-score, P, CAC, GC, and R. For every architecture, cross-validation was also performed, and the predictions were assessed for each method using evaluation methods explained in Table 3.

Our results show that when the training set is augmented 10 times, the performance improves to a greater extent. Hence, the refinement of the proposed post-processing methods is effective. In comparison to Crack-GF, Crack-BN, and SegNet, our proposed architecture shows obvious improvements. It is already reported that the traditional methods involve post-processing (i.e. length constraint, curvature and

geometric features etc.). Therefore, it is indispensable for obtaining continuous and complete thin cracks. However, the convolutional neural networks display this weakness.

10 Conclusion

The manual investigation of damages incurred to infrastructure is a challenging endeavour that is time-consuming and lacks objectivity and reliability. Therefore, automatic crack detection through techniques such as image processing is inevitable, but the influence of noise caused by lighting, blurring and other factors need to be addressed. Amongst the different deep learning approaches CNNs to provide automatic learning of image features instead of image feature extraction thus making it less influenced by noises. For this reason, we suggest a framework based on deep hierarchical CNN architecture along with Cycle GAN for predicting crack segmentation for each pixel in an approach that is end-to-end.

The proposed method utilizes the extended FCN (Fully Convolutional Networks), the DSN (Deeply Supervised Nets) and a U-net architecture. The DSN delivers direct and integrated feature supervision at each convolutional stage. Moreover, the intricately designed model network learns and aggregates features as it moves from the low convolutional layers to the high-level convolutional layers during the training procedure. Thus, the used architecture is different from the ones used traditionally which mainly rely on using the last convolutional layer. Additionally, for the refinement of prediction results, we utilized the Phase shift based guided filtering. Our proposed deep hierarchical convolutional neural network (CNN) architecture achieved advanced/high-tech performances on the considered dataset showing using a GF pipeline displayed the highest-Class Average Accuracy, Mean IOU, Global Accuracy, Recall and Precision values of 0.931, 0.878, 0.989, 0.861, and 0.881 respectively. Several limitations exist in the proposed CNNs framework such as limitations in terms of pixel-perfect accuracy. Other limitations that might be evident are that it requires many computational resources, because of the generative approach. In future, this work can be converted into a knowledge distillation architecture (student and teacher) [38]. Where a complex network (teacher) is used to learn the underlying mapping and at the same time enforce limitations in the complexity of the student model.

Acknowledgements We would like to thank CDRI for their support in conducting this research.

Author Contributions Methodology, HSM and AWAH; investigation, AWAH and STW; writing—original draft preparation, HSM and AWAH; writing—review and editing, AWAH, MRI, and STW;

supervision AWAH and STW. All authors have read and agreed to the published version of the manuscript.

Funding This research received no external funding.

Data Availability All the data sets used and generated in this study along with the codes and models that support our findings are the property of VERIS-Australia.

Declarations

Conflicts of interest The authors declare no conflicts of interest.

Institutional review board statement Not applicable.

Informed consent Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abdel-Qader I, Abudayyeh O, Kelly ME. Analysis of edge-detection techniques for crack identification in bridges. *J Comput Civ Eng.* 2003;17:255–63.
2. Yang G, Liu K, Zhao Z, Zhang J, Chen X, Chen BM. Datasets and methods for boosting infrastructure inspection: a survey on defect classification. In: 2022 IEEE 17th international conference on control & automation (ICCA), 2022. IEEE; pp. 15–22
3. Bai Y, Zha B, Sezen H, Yilmaz A. Engineering deep learning methods on automatic detection of damage in infrastructure due to extreme events. *Struct Health Monit.* 2022. <https://doi.org/10.1177/14759217221083649>.
4. Gal Y, Ghahramani Z. Bayesian convolutional neural networks with Bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.
5. Asgari Taghanaki S, et al. Deep semantic segmentation of natural and medical images: a review. *Artif Intell Rev.* 2021;54:137–78.
6. Gibb S, La HM, Louis S. A genetic algorithm for convolutional network structure optimization for concrete crack detection. In: 2018 IEEE congress on evolutionary computation (CEC), 2018. IEEE; pp. 1–8.
7. Han L, et al. Convective precipitation nowcasting using U-Net model. *IEEE Trans Geosci Remote Sens.* 2021;60:1–8.
8. Hoang ND. Detection of surface crack in building structures using image processing technique with an improved Otsu method for image thresholding. *Adv Civ Eng.* 2018;2018:1–10.
9. Agyemang IO, Zhang X, Acheampong D, Adjei-Mensah I, Kusi GA, Mawuli BC, Agleby BLY. Autonomous health assessment of civil infrastructure using deep learning and smart devices. *Autom Constr.* 2022;141: 104396.

10. Mohtasham Khani M, et al. Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines. *Struct Health Monit.* 2020;19:1440–52.
11. Ali L, Alnajjar F, Khan W, Serhani MA, Al Jassmi H. Bibliometric analysis and review of deep learning-based crack detection literature published between 2010 and 2022. *Buildings.* 2022;12(4):432.
12. Qu Z, et al. Crack detection of concrete pavement with cross-entropy loss function and improved VGG16 network model. *IEEE Access.* 2020;8:54564–73.
13. Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell.* 2017;39:640–51.
14. Sitara SN. Review and analysis of crack detection and classification techniques based on crack types. *Int J Appl Eng Res.* 2018;13:6056–62.
15. Yuan M, Liu Z, Wang F. Using the wide-range attention U-Net for road segmentation. *Remote Sens Lett.* 2019;10:506–15.
16. Liu Y, et al. DeepCrack: a deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing.* 2019;338:139–53.
17. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd International conference on learning representations. In: ICLR 2015—conference track proceedings, 2015.
18. Yu Y-H, Kwok NM, Ha QP. Color tracking for multiple robot control using a system-on-programmable-chip. *Autom Constr.* 2011;20:669–76.
19. Samadani R, Vesecky JF. Finding Curvilinear Features In Speckled Images. *IEEE Trans Geosci Remote Sens.* 1990;28:669–73.
20. Alipour M, Harris DK. Increasing the robustness of material-specific deep learning models for crack detection across different materials. *Eng Struct.* 2020;206: 110157.
21. Cha YJ, Choi W, Büyükoztürk O. Deep learning-based crack damage detection using convolutional neural networks. *Comput Aided Civ Infrastruct Eng.* 2017;32:361–78.
22. Ren Y, et al. Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Constr Build Mater.* 2020;234: 117367.
23. Mohan A, Poobal S. Crack detection using image processing: a critical review and analysis. *Alex Eng J.* 2018;57:787–98.
24. Mou L, Zhu XX. Spatiotemporal scene interpretation of space videos via deep neural network and tracklet analysis. In: 2016 IEEE international geoscience and remote sensing symposium (IGARSS), 2016. IEEE; pp. 1823–1826.
25. Safiuddin M, et al. Early-age cracking in concrete: causes, consequences, remedial measures, and recommendations. *Appl Sci.* 2018;8:1730.
26. Shi W, et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), 2016. IEEE; pp. 1874–1883.
27. Liu Y, et al. Richer convolutional features for edge detection. *IEEE Trans Pattern Anal Mach Intell.* 2019;41:1939–46.
28. Özgenel ÇF, Sorguç AG. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In: ISARC 2018—35th international symposium on automation and robotics in construction and international AEC/FM hackathon: the future of building things, 2018. IAARC Publications; pp. 1–8.
29. Wei Y, Wang Z, Xu M. Road structure refined CNN for road extraction in aerial image. *IEEE Geosci Remote Sens Lett.* 2017;14:709–13.
30. Zhang K, Zhang Y, Cheng HD. Self-supervised structure learning for crack detection based on cycle-consistent generative adversarial networks. *J Comput Civ Eng.* 2020;34:04020004.
31. Su T-C. Application of computer vision to crack detection of concrete structure. *Int J Eng Technol.* 2013;5:457–61.
32. Prasanna P, et al. Automated crack detection on concrete bridges. *IEEE Trans Autom Sci Eng.* 2016;13:591–9.
33. Rana A, et al. Deep tone mapping operator for high dynamic range images. *IEEE Trans Image Process.* 2019;29:1285–98.
34. Stentoumis C, et al. A holistic approach for inspection of civil infrastructures based on computer vision techniques. *ISPRS Int Arch Photogramm Remote Sens Spat Inf Sci.* 2016;XLI-B5:131–8.
35. Zhu J-Y, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE international conference on computer vision (ICCV), 2017. IEEE; pp. 2242–2251.
36. Zheng S et al. Conditional random fields as recurrent neural networks. In: 2015 IEEE international conference on computer vision (ICCV), 2015. IEEE; pp. 1529–1537.
37. Redmon J et al. You only look once: unified, real-time object detection In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2016. pp. 779–788.
38. Gou J, et al. Knowledge distillation: a survey. *Int J Comput Vision.* 2021;129(6):1789–819.

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