sensors

MDPI

Article

Pixel Intensity Resemblance Measurement and Deep Learning Based Computer Vision Model for Crack Detection and Analysis

Nirmala Paramanandham [1,*], Kishore Rajendiran [2], Florence Gnana Poovathy J [1], Yeshwant Santhanakrishnan Premanand [3], Sanjeeve Raveenthiran Mallichetty [4] and Pramod Kumar [1]

[1] School of Electronics Engineering, Vellore Institute of Technology, Chennai Campus, Chennai 600127, India
[2] Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam 603110, India
[3] Department of Software Engineering, Rochester Institute of Technology, GoLisano College of Computing and Information Sciences, Rochester, NY 14623, USA
[4] Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA
* Correspondence: nirmala.p@vit.ac.in; Tel.: +91-770-845-7482

Abstract: This research article is aimed at improving the efficiency of a computer vision system that uses image processing for detecting cracks. Images are prone to noise when captured using drones or under various lighting conditions. To analyze this, the images were gathered under various conditions. To address the noise issue and to classify the cracks based on the severity level, a novel technique is proposed using a pixel-intensity resemblance measurement (PIRM) rule. Using PIRM, the noisy images and noiseless images were classified. Then, the noise was filtered using a median filter. The cracks were detected using VGG-16, ResNet-50 and InceptionResNet-V2 models. Once the crack was detected, the images were then segregated using a crack risk-analysis algorithm. Based on the severity level of the crack, an alert can be given to the authorized person to take the necessary action to avoid major accidents. The proposed technique achieved a 6% improvement without PIRM and a 10% improvement with the PIRM rule for the VGG-16 model. Similarly, it showed 3 and 10% for ResNet-50, 2 and 3% for Inception ResNet and a 9 and 10% increment for the Xception model. When the images were corrupted from a single noise alone, 95.6% accuracy was achieved using the ResNet-50 model for Gaussian noise, 99.65% accuracy was achieved through Inception ResNet-v2 for Poisson noise, and 99.95% accuracy was achieved by the Xception model for speckle noise.

Keywords: cracks; deep learning; detection; images; noise; integrity; safety

check for updates

Citation: Paramanandham, N.; Rajendiran, K.; Poovathy J, F.G.; Premanand, Y.S.; Mallichetty, S.R.; Kumar, P. Pixel Intensity Resemblance Measurement and Deep Learning Based Computer Vision Model for Crack Detection and Analysis. *Sensors* **2023**, *23*, 2954. https://doi.org/10.3390/s23062954

Academic Editors: Yun Zhang, Kwong Tak Wu Sam, Xu Long and Tiesong Zhao

Received: 29 January 2023
Revised: 22 February 2023
Accepted: 23 February 2023
Published: 8 March 2023

1. Introduction

The identification of cracks on different types of structures has always been tedious and time consuming work. Regular checks have to be made in order to prevent any serious damage to infrastructure. Traditional inspections would require the use of specialized personnel to manually check for any cracks. This process is greatly complicated when it has to be done in areas such as roads, bridges and highways. It can cause disturbance to regular work or create traffic due to the need to employ additional platforms or machinery for aiding in the inspection process. Furthermore, after the examination, the reports are usually checked manually to identify the underlying issues. This procedure is time consuming and costly to implement. In order to reduce the cost, time and labor involved in such scenarios, the use of unmanned aerial vehicles (UAV) and transfer learning methods can be used to identify cracks. Once a crack is identified, it can be separated, based on the severity level. This tedious process would be simplified by automation. The main goal of this research article is to improve this nondestructive method of investigation which can be employed at a considerably lower cost, while maintaining good accuracy. The convolutional neural network (CNN) method is one of the most efficient network methods which can be used for image-based crack detection. CNNs have demonstrated much higher accuracy than many image-enhancement algorithms, such as the morphological approach, edge detectors, wavelet analysis, etc. [1].

Over the years, machine learning and convolutional networks have been displaying strong capability for feature extraction and target detection and many researchers have developed and studied various applications for improving the accuracy. Research carried out by Cuong Nguyen Kim et al. [1] concentrated on pre-processing. To improve the accuracy, Aravinda S Rao et al. [2] took a different approach in pre-processing where they divided the image into multiple patches before the images were fed into the CNN-based algorithms. Vishal Mandal et al. [3] focused on detecting the cracks. Similar methods have been implemented in various research, such as in Rupinder Pal Singh et al.'s [4] work, where they used a bilateral filter to remove noise as it helped to preserve edges while reducing noise. Another work carried out by Mukund N Naragund et al. [5] showcased a methodology using wavelet transform along with a bilateral filter to denoise and preserve detail to a high degree.

The main contributions of this research are summarized as follows:

1. A deep learning model for crack detection using image processing for computer vision is proposed;
2. In order to detect whether the image has been affected by noise, a unique technique which uses pixel-intensity resemblance is implemented;
3. A binarization-skeletonization-edge detection (BSE) algorithm is proposed for estimating the width of cracks. Based on the width, the images are segregated into high-risk, medium-risk and low-risk cracks using preset thresholds.

Section 2 elaborates the available literature on crack detection. Section 3 explains the proposed work. Sections 4 and 5 discuss the results and conclusions.

2. Background

Cracks can be detected using basic machine learning algorithms. As opting for deep learning is more fruitful in terms of accuracy and speed, many state-of-the-art techniques have concentrated in these techniques. The research article by Raza Ali et al. [6] surveyed different CNN-based algorithms and stated that Unet was the best performer when compared to Pixelnet, Alexnet, Googlenet and a few other algorithms. V Mandal et al. [3] was able to detect cracks in real time by mounting a camera on the dashboard of a moving car. Y Zhang et al. [7] used the YOLO v3 algorithm as a base and was able to detect the cracks efficiently by using MobileNet for transfer learning and the convolutional block attention model. The authors of [8] carried out research on various CNN-based algorithms, and found that MobileNet yielded the best accuracy for a masonry dataset. J. K. Chow et al. [9] carried out crack detection on concrete images using a convolutional autoencoder and decoders. Zhong qu et al. [10] and Cheng Wang et al. [11] discussed improving accuracy by using only two convolutional layers and the Inception model. SY Wang et al. [12] compared R-CNN-based ResNet, visual geometry group (VGG) and feature pyramid network (FPN). It was concluded that VGG16 took less time and memory to detect cracks, but yielding the lowest accuracy. ResNet-50 gave the highest accuracy but took more time and some extra memory.

The addition of certain pre-processing steps can be of great value; they can make or break an algorithm. A very good example would be the research of Thendral et al. [13], where cracks on railway tracks were collected using a camera on a self-moving vehicle and various pre-processing proceduree were carried out to classify the cracks appropriately. Similarly the research by Zhong Qu et al. [10] proved that, using a simple technique of dividing an image into smaller patches, considerable improvement can be achieved, compared to most of the state-of-the-art deep learning models. CV Dung et al. [14] used a fully convolutional network (FCN) and scanned the dataset for common features on crack images and classified the images. Using the FSM module, UH Billah et al. [15] found the weak features of the dataset and eliminated them. They concatenated the encoder-decoder modules and upscaled the remaining features. This method is particularly useful when a dataset has different types of images. Although it improves the accuracy, this method is highly sensitive to the input data.

The research of Zhang et al. [16] relies heavily on the concept of feature fusion. The crack images are very susceptible to noise. It cannot be guaranteed that all the images can be taken in well-lit conditions. In similar research, the researcher used a multiscale-fusion generative adversarial network (GAN) to improve the quality of the output images while preserving the features of the original images. However, they assumed the noise type to be Gaussian and the variance to be between 0.05 and 0.2 [17,18]. Junmei Zhong et al. [19] took a different approach for reducing the noise; by using orthogonal wavelet transform (OWT), the higher scale levels are preserved, and noise in the lower level is filtered by using minimum mean squared error. Although this noise reduction method was yielding better results, they only reported it for images with Gaussian noise. Ehsan Akbari Sekehravani et al. [20], utilized the Canny algorithm for edge detection. Traditionally Canny is implemented with Gaussian filter, but to counteract any type of noise, the authors utilized a filtering approach. Another unique method is denoising the images using the Wiener filter and detecting cracks by the Otsu method [21]. Kittipat Sriwong et al. [22] and discussed various CNN-based algorithms for efficient crack detection. Even though the technique implemented in [23] was not able to carry out a proper categorization of the crack image, it could detect cracks even on road markings using ResNet-v2 algorithm. By adding feature fusion and network in network (NIN) modules, the edges were highlighted and also prevented the loss of model features, in the meanwhile reducing the time complexity. In [24], to inspect the severity of the crack, the authors used crack magnifier. The deep learning models such as VGG-16, ResNet50 and Inception ResNet-V2 are discussed in [25–27]. Paramanandham et al. [28] discussed about concrete crack detection using various deep learnbing models. Qi Chen et al. [29] used the guided filter approach for the removal of noise and analyzed the characterization of the crack structure using Hessian structures followed by refinement process. The authors achieved around 90% in precision, recall and F1 measurements through the implemented approach. Dawei Li et al. [30] developed a defect detection system for metro tunnel surfaces. Junjie Chen and Donghai Liu [31] proposed a model for detecting damage in the water channel based on super pixel segmentation and classification and achieved an accuracy around 91%. Miguel Carrasco et al. [32] discussed a methodology for measuring the width of cracks using smoothing, filtering, segmentation and estimation. The authors of [33–41] proposed several techniques based on CNN, pyramidal residual network for concrete crack detection, binocular vision system for pipe crack and deformation detection and also analyzed the performance of the techniques. From the literature, it can be identified that the existing techniques for detecting cracks can be classified into two broad domains. One is based on the combination of several networks or concentrating on segmentation of cracks. Hence, the proposed technique concentrated on overcoming the limitations in the detection of cracks even though the images are corrupted or have dissimilar structures. Figure 1 shows the general block diagram for crack detection. Once the images are acquired, the database is created. Before classifying the images into crack and non-crack, pre-processing procedures such as removal of noise, contrast enhancement, change in resolution, etc., can be performed to obtain enhanced results. Once the crack has been detected, it can be assessed through evaluation parameters.

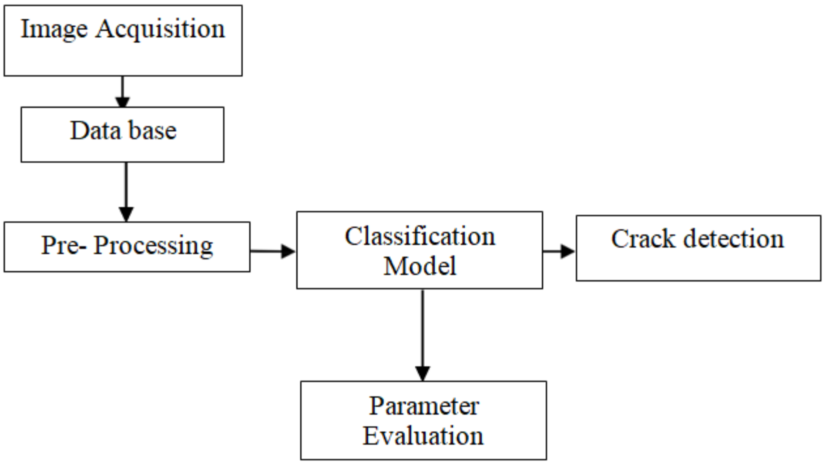


Figure 1. General block diagram for crack detection.

3. Proposed Method

The cracks are detected for both noisy and noiseless environment images captured from various surfaces. To accomplish this, the proposed technique consists of three processes, namely, pixel-intensity resemblance measurement, crack detection using a deep learning model and classification based on the width of the crack. As shown in Figure 2, in the pre-processing stage, the filtering process and the following step (i.e.,) pixel-intensity resemblance algorithm were used for measuring similar pixels. In this pixel-matching technique, the images to be tested are passed through a common filter. The filtered image pixels are compared with the original image. The number of mismatched pixels is calculated and, according to that calculation, the type of noise is determined. Once the type of noise is identified, the proper denoising filters are used for the removal of noise. The filtered images are then segregated properly on the basis of whether the images are inclusive of noise or not. Once the separation filtering of possible noisy images is completed, the images are then passed through a crack-detection model. The images in which cracks have been detected are then passed through the last stage of the algorithm where the width of the crack is determined; by doing so, it segregates the various cracked images into three different categories based on the severity level, namely, high, medium and low so that the appropriate actions can be taken without any delay.

To examine the efficiency of the implemented technique under several noise conditions, images were generated with different noises with Gaussian, salt and pepper, and speckle with various mean and variance levels. The Gaussian noise model [37] is expressed in Equation (1).

$$\mathrm{P(g)} = \sqrt{\frac{1}{2\pi\sigma^2}}\mathrm{e}^{-\frac{(\mathrm{g}-\mu)^2}{2\sigma^2}} \tag{1}$$

where σ denotes the standard deviation, g indicates the gray value and μ represents the mean value.

The binary noise is also called impulse noise and salt and pepper, as its value is either 0 or 255. Speckle noise is also termed as multiplicative noise [37]. It occurs in the same way in an image as Gaussian noise. It is expressed in Equation (2).

$$\mathrm{F(x)} = \frac{\mathrm{x}^{\alpha-1}\mathrm{e}^{\frac{-\mathrm{x}}{\mathrm{a}}}}{(\alpha-1)!\mathrm{a}^{\alpha}} \tag{2}$$

The proposed crack detection model was developed in view of the following parameters:

1. Cracks should be detected on any surfaces captured from any device under any environment;
2. Time complexity is considered;
3. Once a crack is identified, it should be categorized and an immediate alert will be given to the authority in order to avoid major accidents.

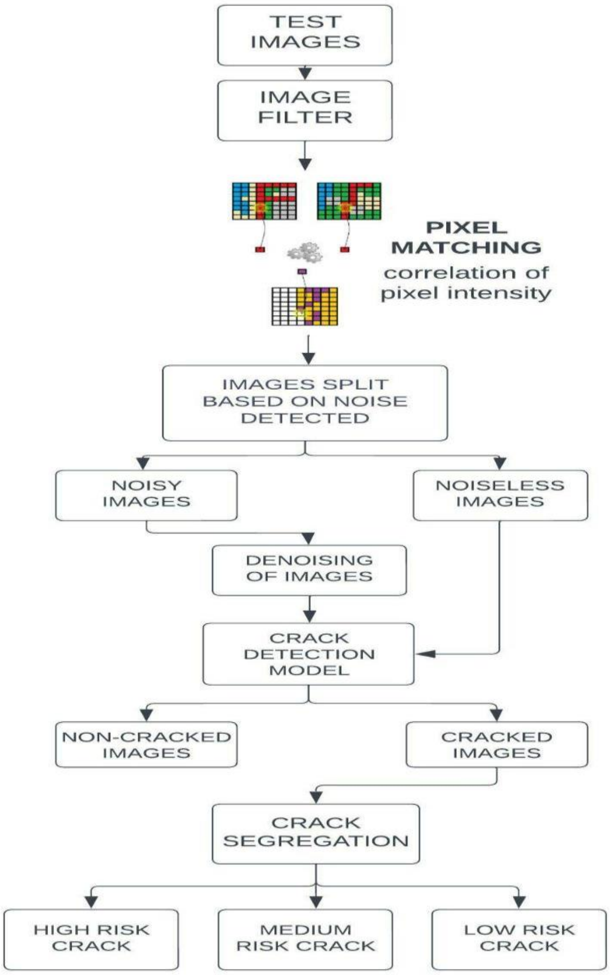


Figure 2. Flowchart of the proposed algorithm.

3.1. Filtering and Pixel-Intensity Resemblance Measurement for Noise Classification

The images of any structure or surface taken from a drone or some other device are classified into noisy and noiseless (very little noise) images, based on the measurement of pixel-intensity resemblance and a filter-based approach. To classify the images accurately, the images are initially passed through a common filter. The filtered image pixels are compared to the database images for finding similarities between the pixels. After extensive study of filters and from Table 1, it was found that the median filter yielded better results when compared to all other filters for the proposed technique. Hence, all the images were passed through the median filter and the filtered image was then given to the next stage. A median filter is a non-linear digital filtering technique that is often used in the pre-processing of images as it helps remove the noise efficiently but preserves the edge details. It is very useful for edge detection and other image-based detection methods.

Table 1. Noise estimation through filters with PIRM.

Image Filters	Classification Accuracy (%)
Mean Filter	87.3
Median Filter	91.5
Low Pass Filter	83.3
Gaussian Filter	88.1

The filtered images are passed through the pixel-matching algorithm where the pixels of the original image and the filtered image are compared and the number of matched and mismatched pixels are computed. The code works by extracting the intensity of pixels that have the same coordinates within the image. If the pixel intensity of both images matches, then it is marked as a common pixel, otherwise it is denoted as a mismatched pixel. In Figure 3, the yellow output is the mismatched pixels while the purple output is when the pixels match.

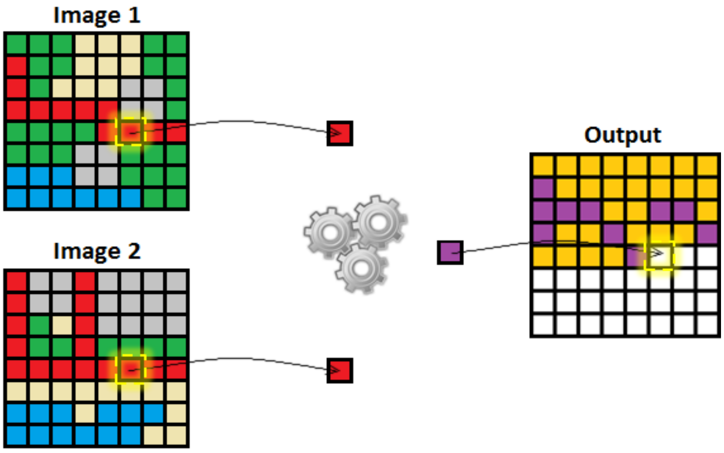


Figure 3. Visual representation of pixel matching process.

For each image, the mismatches are compared against certain thresholds and the decision is made whether to use the original image or the filtered image. If the number of mismatched pixels is between 15 and 100 then the original image is passed through to the deep learning model, as the filtering of images with supposedly very little or no noise ends in needless loss of the image. The images whose pixel mismatch range is over 100 are deemed to be noisy, the original image is first filtered and then passed through the detection algorithm.

3.2. Noise Estimation

In order to decide the filter that should be used for denoising, the level of noise is estimated and the flow for the estimation is explained using the Equations (3)–(8). Noise is estimated for various types of noises with different mean and variance levels. Let us consider an image I with the patch size p, row R and column C and dataset D that is specified in Equations (3) and (4)

$$\mathrm{I} \,\epsilon\, \mathrm{A}^{\mathrm{RXCX3}}, \tag{3}$$

and

$$\mathrm{D} = \{\mathrm{d_i}\}_{\mathrm{i}=1}^{\mathrm{s}} \tag{4}$$

where D contains s = (R − d + 1) (C − d + 1) patches with size q = 3p^2

$$\mu = \sum_{\mathrm{i}=1}^{\mathrm{s}} \mathrm{x_i}, \tag{5}$$

$$\text{Covariance Matrix } \sum = \frac{1}{\mathrm{s}} \sum_{\mathrm{i}=1}^{\mathrm{s}} (\mathrm{x_i} - \mu)(\mathrm{x_i} - \mu)^{\mathrm{t}}\,. \tag{6}$$

Computing the Eigen values $\{\lambda_i\}_{j=1}^{q}$ of the covariance matrix Σ with q = p^2 and order $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r$

For j = 1: q, median τ is calculated

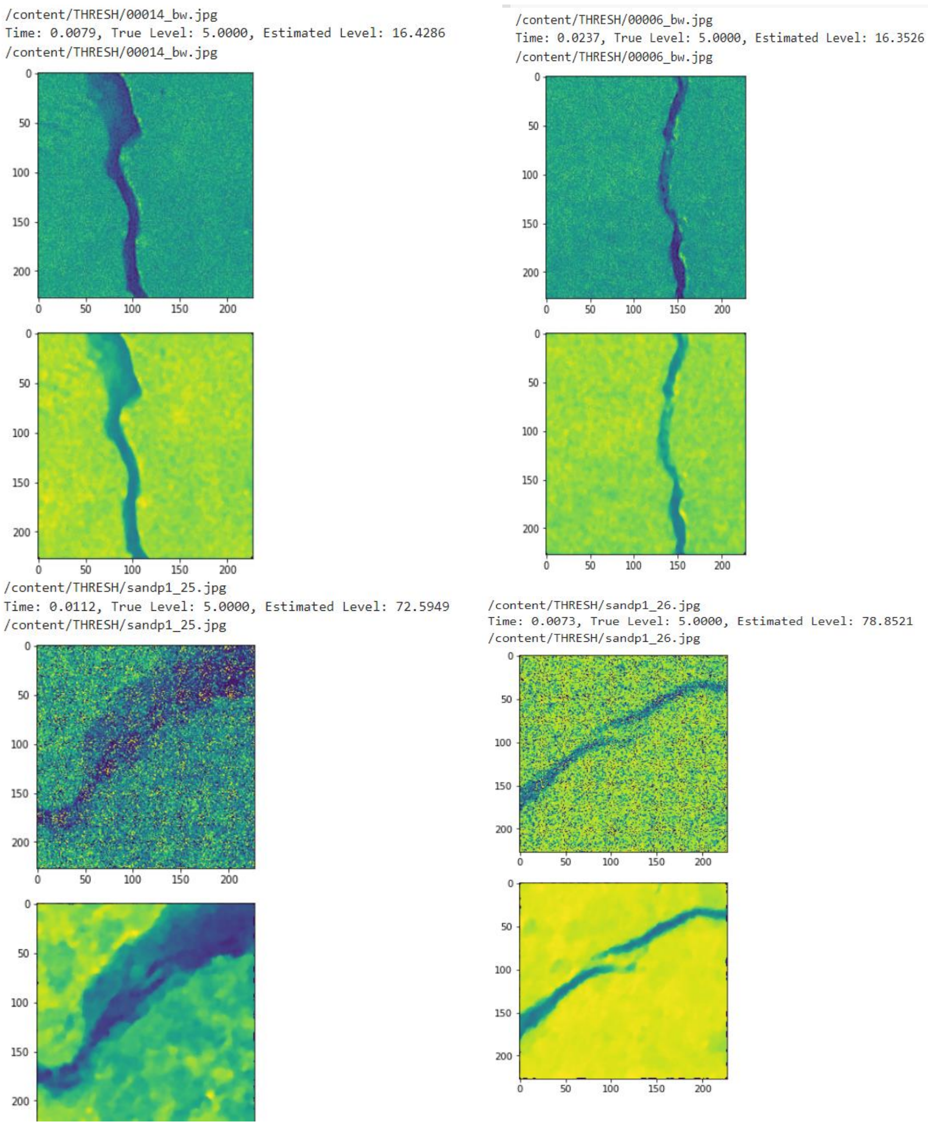
$$\tau = \frac{1}{\mathrm{q} - \mathrm{t} + 1} \sum_{\mathrm{k}=1}^{\mathrm{q}} \lambda_{\mathrm{k}}\,, \tag{7}$$

If τ is the median of the set $\{\lambda_k\}_{k=1}^{q}$ then,

$$\sigma = \sqrt{\tau}\,. \tag{8}$$

where σ represents the estimated noise level.

If the estimated range is within ±5, it is lying under Gaussian noise and the Wiener filter is chosen for denoising these types of images as it is more appropriate. Similarly, an image with the estimated range 15 to 25 specifies speckle noise, the mean filter is used for denoising and if it is greater than 30, it shows the image is corrupted due to salt and pepper. If the images are corrupted by salt and pepper noise, the median filter is used for the removal of noise. Figure 4 shows the noise estimation of the proposed technique.



```
/content/THRESH/00014_bw.jpg
Time: 0.0079, True Level: 5.0000, Estimated Level: 16.4286
/content/THRESH/00014_bw.jpg
```

```
/content/THRESH/00006_bw.jpg
Time: 0.0237, True Level: 5.0000, Estimated Level: 16.3526
/content/THRESH/00006_bw.jpg
```

```
/content/THRESH/sandp1_25.jpg
Time: 0.0112, True Level: 5.0000, Estimated Level: 72.5949
/content/THRESH/sandp1_25.jpg
```

```
/content/THRESH/sandp1_26.jpg
Time: 0.0073, True Level: 5.0000, Estimated Level: 78.8521
/content/THRESH/sandp1_26.jpg
```

Figure 4. Noise estimation.

3.3. Deep Learning Models Used for Crack Detection

3.3.1. VGG-16 Architecture

The VGG-16 consists of 13 convolutional layers and three fully connected layers as shown in Figure 5. A set of filters comprises a convolutional layer which is an essential block in any convolutional neural network. VGG-16 [25] has 13 of them. The parameters of the filters have to be learned. The size of the filter must be relatively less than the input. The features of the training set are extracted only using convolutional layers. The next layer in VGG-16 is the pooling layer. Generally, pooling layers are added between two convolutional layers. Pooling layers reduce the number of parameters between successive layers. There are two pooling functions, namely average and max pooling. Max pooling is generally preferred as it functions more efficiently. The flattened layer in VGG-16 converts feature maps into 1D tensors. The last layer is the fully connected layer which gives the output of the model.

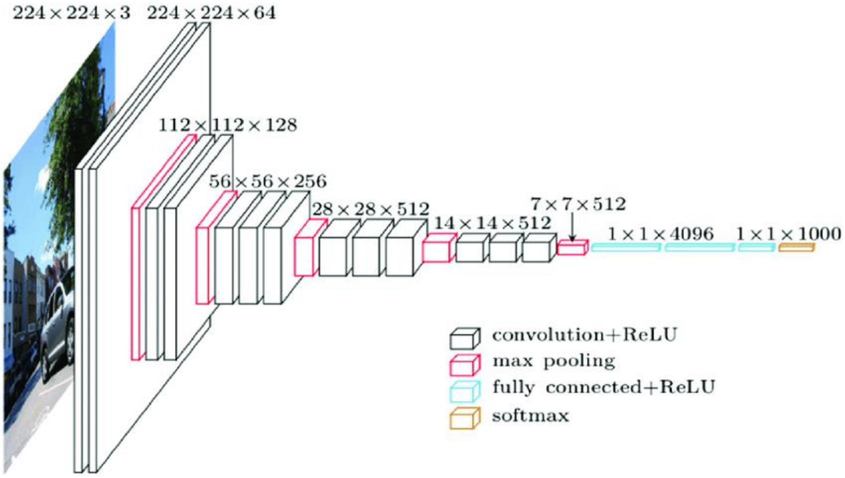


Figure 5. Architecture of VGG-16.

3.3.2. ResNet-50 Architecture

ResNet-50 is a variant of ResNet with 50 neural network layers [26] as shown in Figure 6, redrawn from [39]. Over the years, the higher accuracy and efficiency of neural network models have been achieved by deepening the neural network model, i.e., adding more layers and blocks or changing the filter size. This, however, is not always the case. Adding more and more layers can also cause performance degradation in deep learning. In order to overcome this, residual networks which are made up of residual blocks have been invented. The concept of skip connection is being introduced in residual models. While training a model, the skip connections skip some of the layers in the model (layers that are skipped vary from model to model). The output of one layer is fed as the input to another layer. This basically solves the problem of vanishing gradients in deep neural networks. The skip connections also ensure that the higher layers and lower layers of a model perform efficiently. The residual blocks in the model help to increase efficiency as learning becomes much easier.

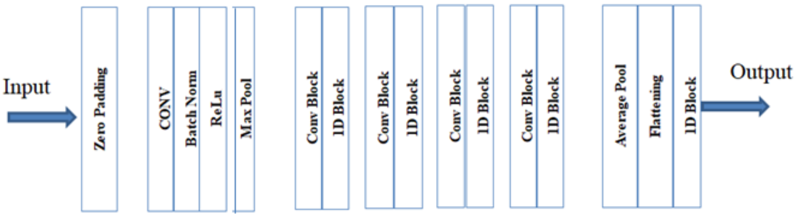


Figure 6. The architecture of ResNet-50.

3.3.3. Inception ResNet-V2

Inception ResNet-v2 basically uses the inception architecture combined with the residual connections from the ResNet network. The major improvement from the traditional model is the addition of a filter expansion layer to scale up the dimensionality of the filter bank before the addition to match the depth of input. The network has a total of 164 layers, as shown in Figure 7, redrawn from [39], and can classify the images in up to 1000 different categories, in the same way as the VGG-16 and ResNet-50. The input size of this network is 299 × 299 and the output is a list of estimated class probabilities.

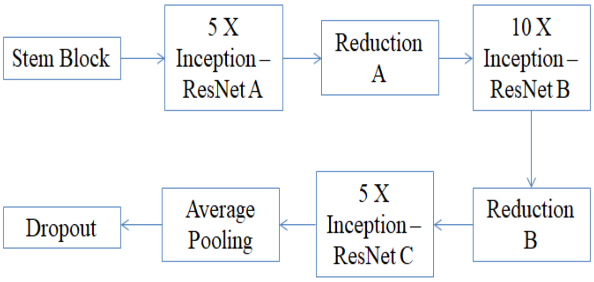
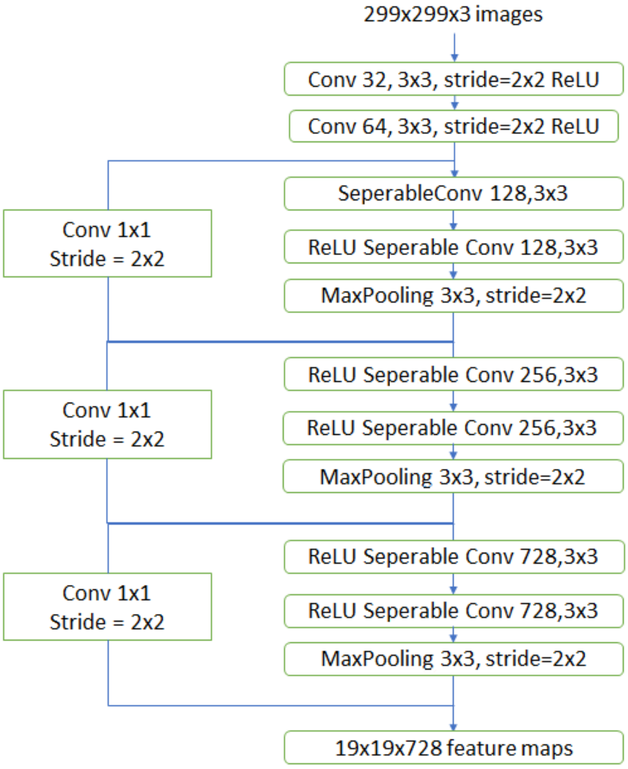


Figure 7. The architecture of Inception ResNet-v2.

3.3.4. Xception Model

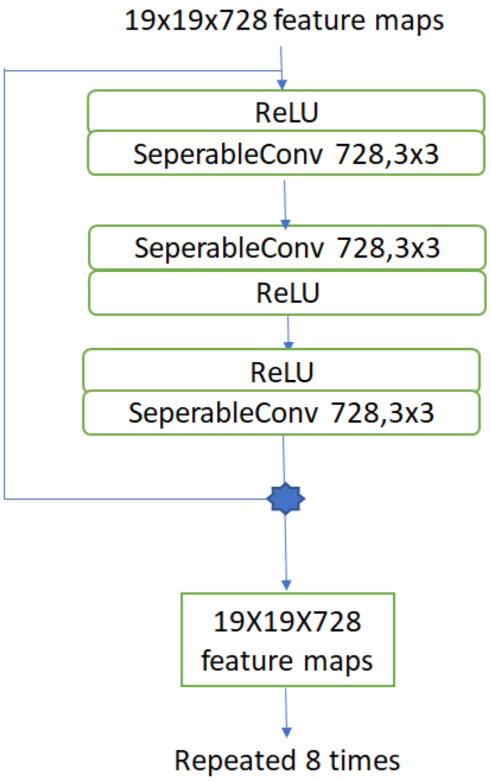
The Xception model uses depth-wise separable convolutions and works as shown in Figure 8a–c, redrawn from [27]. A general convolution step makes the spatial-wise and channel-wise computation in one single step. However, on the other hand, the depth-wise separable convolution divides the process of computation into two different steps. The depth-wise convolution initially adds a single convolutional filter to each input channel. It is then followed by point-wise convolution which creates a linear combination of the output from the depth-wise convolution. This method improves the efficiency of the model.

The word "Xception" literally translates as "extreme inception". It basically means that the properties of the inception model are extremized to give better results. In the traditional inception neural network model, the original input image was compressed using a one-by-one convolution. After this, different types of filters were used on each depth space. However, in the Xception model, this step is reversed. Here, the filter is applied in the first step of the depth map and then the compression of the input takes place. This technique is called depth-wise convolution. The Xception model also does not introduce non-linearity which was the case in the inception model. This is also yet another difference between the models.

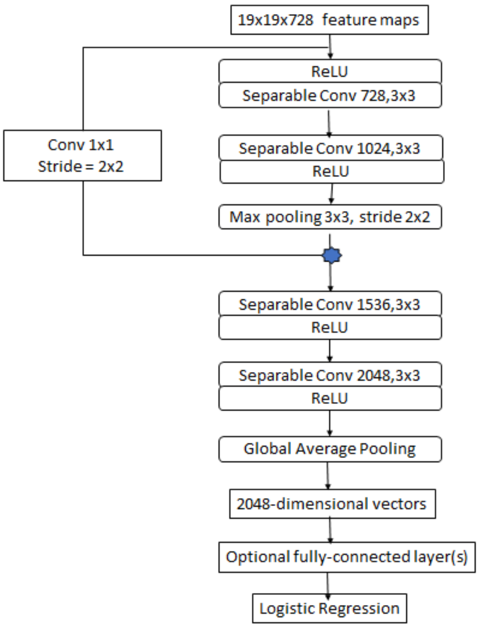

299x299x3 images
Conv 32, 3x3, stride=2x2 ReLU
Conv 64, 3x3, stride=2x2 ReLU
SeperableConv 128,3x3
ReLU Seperable Conv 128,3x3
MaxPooling 3x3, stride=2x2
Conv 1x1 Stride = 2x2
ReLU Seperable Conv 256,3x3
ReLU Seperable Conv 256,3x3
MaxPooling 3x3, stride=2x2
Conv 1x1 Stride = 2x2
ReLU Seperable Conv 728,3x3
ReLU Seperable Conv 728,3x3
MaxPooling 3x3, stride=2x2
Conv 1x1 Stride = 2x2
19x19x728 feature maps

(**a**)

Figure 8. *Cont.*



(**b**)



(**c**)

Figure 8. (**a**) The architecture of Xception entry flow; (**b**) the architecture of Xception middle flow; (**c**) the architecture of Xception exit flow.

3.4. Crack Segregation Based on BSE Algorithm

The images that are identified with cracks are passed to the proposed crack risk analysis algorithm (binarization-skeletonization-edge detection—BSE) where the width of the crack is estimated. Based on the width, the images are segregated into high-risk, medium-risk and low-risk cracks by the preset threshold.

Crack risk analysis using BSE algorithm:

1. Image binarization: this is the operation of dividing the image into black/white pixels in order to separate the cracks and non-cracks within the image;
2. Skeletonization: extracts the central skeleton of the crack which helps to identify the progression of the crack. Hence, it is possible to find the crack width by drawing a line perpendicular to the crack propagation direction at the pixel on the skeleton;
3. Edge detection: extracts the outline of the crack. From the skeleton, the line perpendicular to the crack propagation direction and the crack outline are used together to find the crack width.

4. Results and Discussion

Once the type of noise is estimated, the appropriate filters are applied. These images are converted into gray scale images and the width is calculated. The crack estimation accuracy is drastically increased when the images are denoised based on the proposed technique. For calculating performance evaluation parameters, confusion matrix is used and it is represented in Table 2. Tables 3 and 4 show the accuracy of various deep learning models before and after denoising. From Tables 3 and 4, it is proved that the proposed technique is efficient in denoising and detecting the cracks. Figure 9 shows the visual representation of crack width prediction. Once the crack was identified, it was classified into high risk, medium risk and low risk and these are shown in Figures 10–14. The high-risk cracks would be immediately alerted to the authority to ensure avoidance of major disasters or accidents and to prevent a calamity. The work was implemented using Python software in Google Colab. The proposed system will be very helpful to many industries and public transport authorities, including bridges on the pathway. To assess the effectiveness of the proposed technique, it was compared with state-of-the-art techniques such as Auto-CAE [9], ResNet-50 [26], Crack Hessian [29] and Seg + SVM [30] and the results are tabulated in Table 5.

Table 2. Confusion matrix.

	Predicted No	**Predicted Yes**
Actual No	TN	FP
Actual Yes	FN	TP

Table 3. Accuracy with and without noise.

NOISE	VGG-16	RESNET-50	INCEPTION RESNET-V2	XCEPTION
No noise	99.9%	98%	99.98%	99.95%
Salt and pepper	50%	50%	56.05%	50%
Gaussian	50%	89.05%	56.15%	50%
Poisson	99.1%	99.2%	98.75%	99.25%
Speckle	96.05%	91.9%	99.55%	99.7%
All noises	73.79%	82.54%	85.47%	74.74%
All noises + no noise	79.41%	85.63%	88.36%	79.79%

Table 4. Accuracy of various models using proposed technique.

NOISE	VGG-16	RESNET-50	INCEPTION RESNET-V2	XCEPTION
Salt and pepper	81%	88.7%	96.3%	99.65%
Gaussian	50%	95.6%	90.95%	50%
Poisson	99.1%	99.2%	99.65%	99.25%
Speckle	99.2%	98.2%	99.85%	99.95%
All noises	82.25%	86.25%	87.89%	87.23%
All noises + no noise	85.79%	88.32%	90.3%	88.74%
All noises + no noise **(with PIRM)**	89.56%	95.78%	90.9%	89.57%

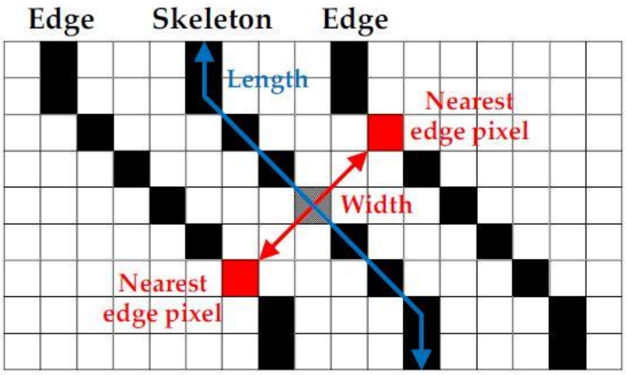


Figure 9. Visual representation of predicting the crack width.

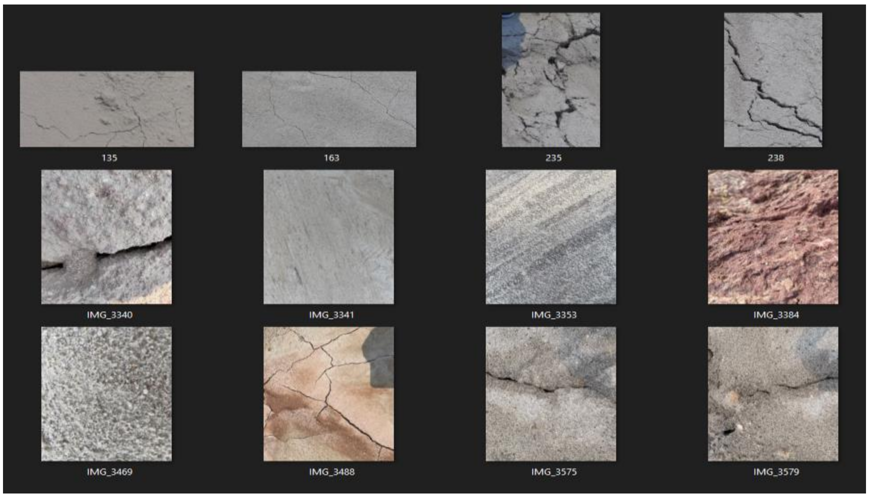
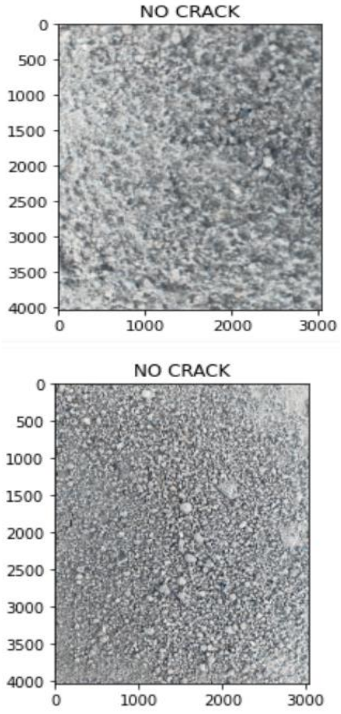
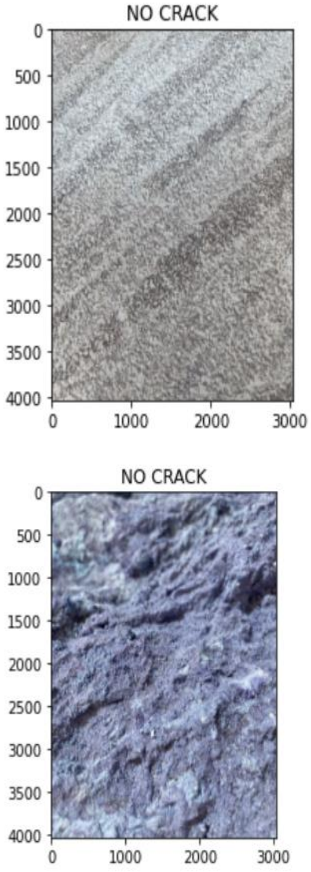


Figure 10. Real-time sample crack images used in the experiment.



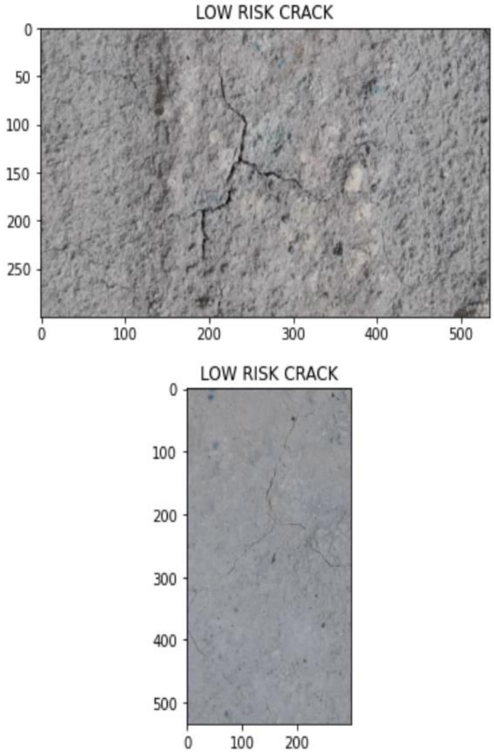
(**a**)

Figure 11. *Cont.*



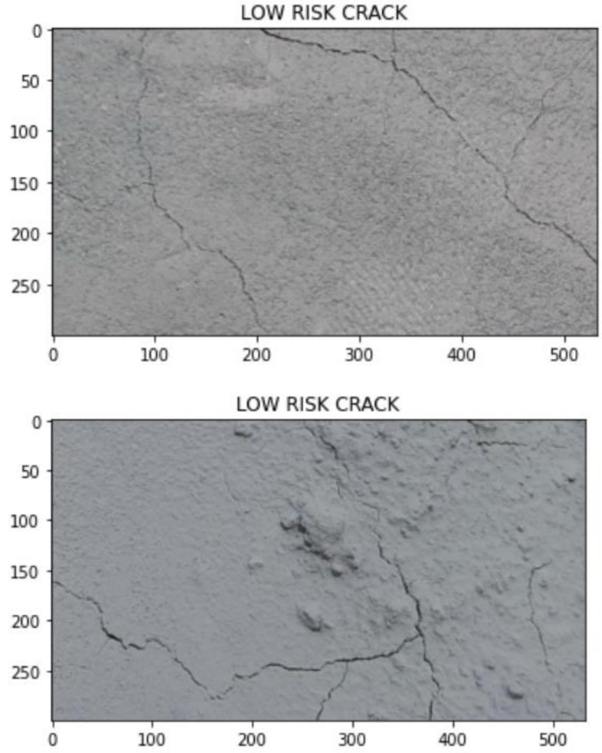
(**b**)

Figure 11. (**a**,**b**) Sample output images from the model classified as "No Crack".



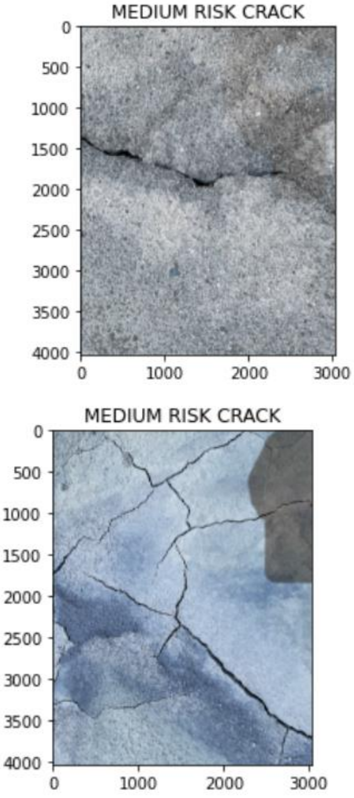
(**a**)

Figure 12. *Cont*.



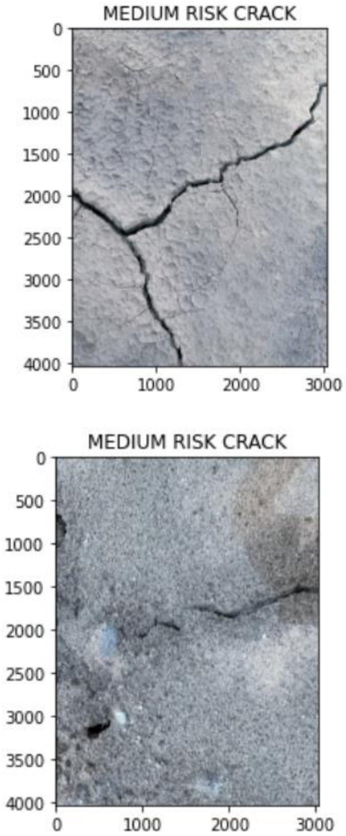
(**b**)

Figure 12. (**a**,**b**) Sample output images from the model classified as "Low-Risk Crack" for field images.



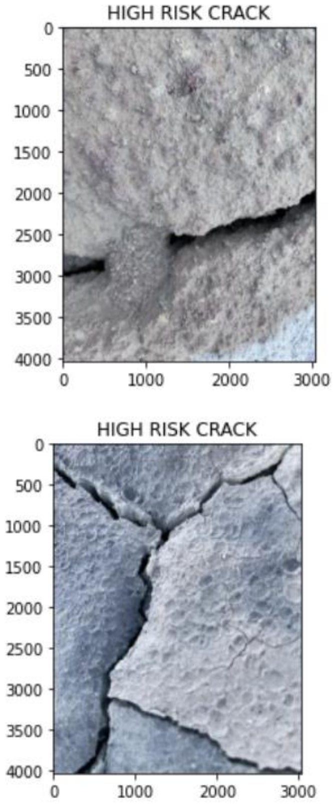
(**a**)

Figure 13. *Cont.*



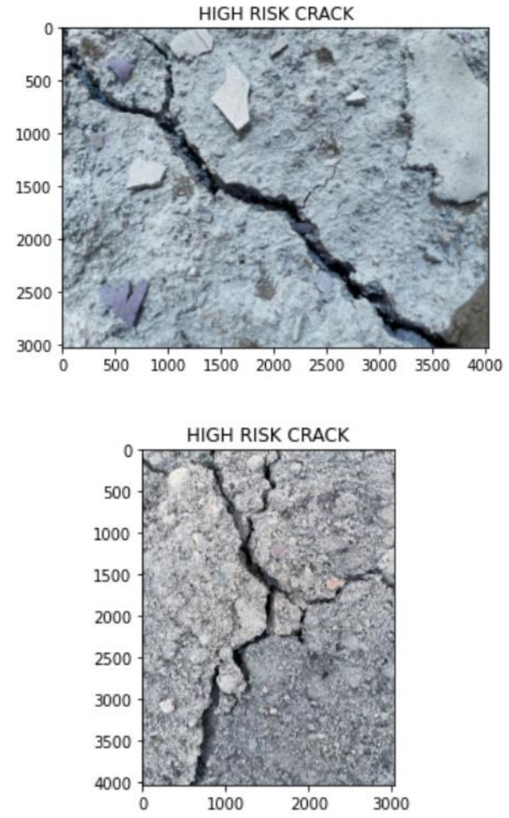
(**b**)

Figure 13. (**a**,**b**) Sample output images from the model classified as "Medium-Risk Crack".



(**a**)

Figure 14. *Cont.*



(**b**)

Figure 14. (**a**,**b**) Sample output images from the model classified as "High-Risk Crack".

Table 5. Comparison with the state-of-the-art techniques.

Techniques	Accuracy	Specificity	Recall	Precision	F1 Score
ResNet-50 [26]	88.36	89.06	87.16	89.46	88.11
Auto-CAE [9]	89.05	89.95	87.32	90.05	88.75
Crack Hessian [29]	91.2	91.9	90.03	91.72	90.9
Seg+ SVM [31]	91.7	91.05	90.27	91.35	91.23
Proposed- (PIRM + BSE)	**95.78**	**96.48**	**94.38**	**96.18**	**95.58**

By searching the skeleton of the image through breadth-first search (BFS), the direction of the crack was estimated. Then the distance was calculated when the line of perpendicular met the edge of the crack, as shown in Figure 9. This was repeated multiple times until various widths were covered and the average of the distances obtained was used as the estimated width value of the crack. Figure 10 shows some sample images collected from industry. With multiple hand-selected images of various degrees of severity in the crack, accurate thresholds for high, medium, and low risk were identified by the proposed model. Figure 11a,b shows the sample of predicted output of the model "No Crack", even though the images have many irregularities, grainy surface and complicated structure. Figure 12a,b shows the predicted output of the model "Low-Risk Crack", Figure 13a,b shows the classified output of "Medium-Risk Crack". Once a high-risk crack is detected, an immediate alert will be given to the authority and the necessary action will be taken to avoid accidents. The high-risk-classified crack images are shown in Figure 14a,b. Even though the images have different surface properties, the proposed model can effectively classify according to category. A confusion matrix was generated for all the models to assess the efficiency of the models and this is given in Table 2.

This table indicates the predictions made by the model and how right/wrong those predictions were. The parameters such as accuracy, precision, recall and F1 score were calculated using Equations (9)–(12), respectively.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{9}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{10}$$

$$\text{F1 Score}: \ 2 \text{ X } \frac{\text{Precision X Recall}}{\text{Precision} + \text{Recall}}, \tag{11}$$

and

$$\text{Accuracy}: \ \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}. \tag{12}$$

5. Discussion and Conclusions

The adverse effects of noise on image-based detection methodologies, especially in the detection of cracks, are successfully identified in this article by exploring various deep learning algorithms. The discussed deep learning models showed around 30–50% decrease in accuracy when the test images were noisy. To counteract this, the noise was estimated and the appropriate filters were used for denoising using the developed technique. From the results, it was identified that the implemented technique had different effects on the various models. When the dataset contained all the types of images excluding the images corrupted from Gaussian noise, speckle noise, Poisson noise, salt and pepper noise, the proposed technique achieved 6% improvement without PIRM and 10% improvement with the PIRM rule for the VGG-16 model. Similarly, it showed a 3 and 10% improvement for ResNet-50, a 2 and 3% improvement for Inception ResNet and a 9 and 10% improvement for the Xception model. When the images were corrupted from single noise, 95.6% accuracy was achieved using the ResNet-50 model for Gaussian noise, 99.65% accuracy was achieved through Inception ResNet-v2 for Poisson noise, and 99.95% accuracy was achieved by the Xception model for speckle noise.

From these results, it was concluded that the ResNet-50 model was the most suitable both when the test images contained no noise, as well as for all types of noisy images, achieving 95.78% with the proposed technique. To evaluate the performance of the developed technique, it was compared with state-of-the-art techniques and the obtained results depicted that the proposed technique outperformed the existing techniques. Thus, it can be concluded that the pixel-intensity resemblance measurement, noise estimation and crack classification-based technique proposed here is most suitable for all types of real-time images taken from any environment.

In future, the authors plan to work on the limitations of the proposed work, i.e., detecting cracks and uneven surfaces occurring in various materials such as steel, iron, and compound cylindrical structures due to strain or some other external environmental factors.

Author Contributions: Conceptualization: N.P.; Methodology: Y.S.P., S.R.M., P.K.; Software: Y.S.P., S.R.M., P.K.; Validation: N.P., K.R. and F.G.P.J.; Investigation: K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study did not require ethical approval.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Publicly available data is used and cited appropriately.

Acknowledgments: The authors thank VIT for providing “VIT RGEMS SEED GRANT” for carrying out this research work.

Conflicts of Interest: We have no conflict of interest to disclose.

References

1. Kim, C.N.; Kawamura, K.; Nakamura, H.; Tarighat, A. Research on Automatic Crack Detection for Concrete Infrastructures Using Image Processing and Deep Learning. *Curr. Approaches Sci. Technol. Res.* **2021**, *3*, 46–55.
2. Rao, A.S.; Nguyen, T.; Palaniswami, M.; Ngo, T. Vision-based automated crack detection using convolutional neural networks for condition assessment of infrastructure. *Struct. Health Monit.* **2020**, *20*, 2124–2142. [CrossRef]
3. Mandal, V.; Uong, L.; Adu-Gyamfi, Y. Automated road crack detection using deep convolutional neural networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 5212–5215.
4. Singh, R.P.; Varma, V.; Chaudhary, P. A Hybrid Technique for Medical Image Denoising using NN, Bilateral filter and LDA. *IJFRS* **2012**, *1*, 1–5.
5. Naragund, M.N.; Jagadale, B.N.; Priya, B.S.; Panchaxri, V.H. An Efficient Image Denoising Method based on Bilateral filter Model and Neighshrink SURE. *Int. J. Recent Technol. Eng.* **2019**, *8*, 8470–8475. [CrossRef]
6. Ali, R.; Chuah, J.H.; Abu Talip, M.S.; Mokhtar, N.; Shoaib, M.A. Structural crack detection using deep convolutional neural networks. *Autom. Constr.* **2021**, *133*, 103989. [CrossRef]
7. Zhang, Y.; Huang, J.; Cai, F. On Bridge Surface Crack Detection Based on an Improved YOLO v3 Algorithm. *IFAC-PapersOnLine* **2020**, *53*, 8205–8210. [CrossRef]
8. Li, L.; Zheng, S.; Wang, C.; Zhao, S.; Chai, X.; Peng, L.; Tong, Q.; Wang, J. Crack Detection Method of Sleeper Based on Cascade Convolutional Neural Network. *J. Adv. Transp.* **2022**, *2022*, 1–14. [CrossRef]
9. Chow, J.; Su, Z.; Wu, J.; Tan, P.; Mao, X.; Wang, Y. Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Adv. Eng. Inform.* **2020**, *45*, 101105. [CrossRef]
10. Qu, Z.; Mei, J.; Liu, L.; Zhou, D.-Y. Crack Detection of Concrete Pavement With Cross-Entropy Loss Function and Improved VGG16 Network Model. *IEEE Access* **2020**, *8*, 54564–54573. [CrossRef]
11. Wang, C.; Chen, D.; Hao, L.; Liu, X.; Zeng, Y.; Chen, J.; Zhang, G. Pulmonary image classification based on inception-v3 transfer learning model. *IEEE Access* **2019**, *7*, 146533–146541. [CrossRef]
12. Wang, S.; Guo, T. Transfer Learning-Based Algorithms for the Detection of Fatigue Crack Initiation Sites: A Comparative Study. *Front. Mater.* **2021**, *8*, 756798. [CrossRef]
13. Thendral, R.; Ranjeeth, A. Computer Vision System for Railway Track Crack Detection using Deep Learning Neural Network. In Proceedings of the 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Tamil Nadu, India, 13 May 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 193–196.
14. Dung, C.V.; Anh, L.D. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2018**, *99*, 52–58. [CrossRef]
15. Billah, U.H. Automatic Concrete Defect Identification by Silencing Features of Deep Neural Network. Ph.D. Thesis, University of Nevada, Reno, Nevada, August 2020.
16. Zhang, Q.; Barri, K.; Babanajad, S.K.; Alavi, A.H. Real-Time Detection of Cracks on Concrete Bridge Decks Using Deep Learning in the Frequency Domain. *Engineering* **2020**, *7*, 1786–1796. [CrossRef]
17. Ahmadi, R.; Farahani, J.K.; Sotudeh, F.; Zhaleh, A.; Garshasbi, S. Survey of image denoising techniques. *Life Sci. J.* **2013**, *10*, 753–755.
18. Li, S.; Qian, P.; Zhang, X.; Chen, A. Research on Image Denoising and Super-Resolution Reconstruction Technology of Multiscale-Fusion Images. *Mob. Inf. Syst.* **2021**, *2021*, 1–11. [CrossRef]
19. Zhong, J.; Sun, H. Edge-Preserving Image Denoising Based on Orthogonal Wavelet Transform and Level Sets. *J. Image Graph.* **2018**, *6*, 145–151. [CrossRef]
20. Sekehravani, E.A.; Babulak, E. Implementing Canny Edge Detection Algorithm for Noisy Image. *Bull. Electr. Eng. Inform.* **2020**, *9*, 1404–1410. [CrossRef]
21. Scholar, P.G. Review and analysis of crack detection and classification techniques based on crack types. *Int. J. Appl. Eng. Res.* **2018**, *13*, 6056–6062. [CrossRef]
22. Sriwong, K.; Kerdprasop, K.; Kerdprasop, N. The Study of Noise Effect on CNN-Based Deep Learning from Medical Images. *Int. J. Mach. Learn. Comput.* **2021**, *11*, 202–207. [CrossRef]
23. Wang, J.; He, X.; Faming, S.; Lu, G.; Cong, H.; Jiang, Q. A Real-Time Bridge Crack Detection Method Based on an Improved Inception-Resnet-v2 Structure. *IEEE Access* **2021**, *9*, 93209–93223. [CrossRef]
24. Avendaño, J.C. Identification and Quantification of Concrete Cracks Using Image Analysis and Machine Learning. Master's Thesis, KTH, School of Architecture and the Built Environment (ABE), Civil and Architectural Engineering, Structural Engineering and Bridges, June 2020.
25. Nash, W.; Drummond, T.; Birbilis, N. A review of deep learning in the study of materials degradation. *NPJ Mater. Degrad.* **2018**, *2*, 1–2. [CrossRef]
26. Tang, Z.; Li, M.; Wang, X. Mapping Tea Plantations from VHR Images Using OBIA and Convolutional Neural Networks. *Remote Sens.* **2020**, *12*, 2935. [CrossRef]
27. Boer, M.J.; Vos, R.A. Taxonomic classification of ants (Formicidae) from images using deep learning. *bioRxiv* **2018**, *1*, 407452.
28. Paramanandham, N.; Koppad, D.; Anbalagan, S. Vision Based Crack Detection in Concrete Structures Using Cutting-Edge Deep Learning Techniques. *Trait. Signal* **2022**, *39*, 485–492. [CrossRef]

29. Chen, Q.; Huang, Y.; Sun, H.; Huang, W. Pavement crack detection using hessian structure propagation. *Adv. Eng. Inform.* **2021**, *49*, 101303. [CrossRef]
30. Li, D.; Xie, Q.; Gong, X.; Yu, Z.; Xu, J.; Sun, Y.; Wang, J. Automatic defect detection of metro tunnel surfaces using a vision-based inspection system. *Adv. Eng. Inform.* **2020**, *47*, 101206. [CrossRef]
31. Chen, J.; Liu, D. Bottom-up image detection of water channel slope damages based on superpixel segmentation and support vector machine. *Adv. Eng. Inform.* **2020**, *47*, 101205. [CrossRef]
32. Carrasco, M.; Araya-Letelier, G.; Velázquez, R.; Visconti, P. Image-Based Automated Width Measurement of Surface Cracking. *Sensors* **2021**, *21*, 7534. [CrossRef]
33. An, Q.; Chen, X.; Wang, H.; Yang, H.; Yang, Y.; Huang, W.; Wang, L. Segmentation of Concrete Cracks by Using Fractal Dimension and UHK-Net. *Fractal Fract.* **2022**, *6*, 95. [CrossRef]
34. Fan, Z.; Lin, H.; Li, C.; Su, J.; Bruno, S.; Loprencipe, G. Use of Parallel ResNet for High-Performance Pavement Crack Detection and Measurement. *Sustainability* **2022**, *14*, 1825. [CrossRef]
35. Xu, X.; Zhao, M.; Shi, P.; Ren, R.; He, X.; Wei, X.; Yang, H. Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN. *Sensors* **2022**, *22*, 1215. [CrossRef] [PubMed]
36. Hamishebahar, Y.; Guan, H.; So, S.; Jo, J. A Comprehensive Review of Deep Learning-Based Crack Detection Approaches. *Appl. Sci.* **2022**, *12*, 1374. [CrossRef]
37. Paramanandham, N.; Rajendiran, K. Swarm intelligence based image fusion for noisy images using consecutive pixel intensity. *Multimedia Tools Appl.* **2018**, *77*, 32133–32151. [CrossRef]
38. Nguyen, L.D.; Lin, D.; Lin, Z.; Cao, J. Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–5. [CrossRef]
39. Mukherjee, S. The Annotated ResNet-50. Towards Data Science. 2022. Available online: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758 (accessed on 18 November 2022).
40. Tang, Y.; Zhu, M.; Chen, Z.; Wu, C.; Chen, B.; Li, C.; Li, L. Seismic performance evaluation of recycled aggregate concrete-filled steel tubular columns with field strain detected via a novel mark-free vision method. *Structures* **2022**, *37*, 426–441. [CrossRef]
41. Tan, H.; Dong, S. Pixel-Level Concrete Crack Segmentation Using Pyramidal Residual Network with Omni-Dimensional Dynamic Convolution. *Processes* **2023**, *11*, 546. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.