Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 27 No. 4 (2024)

Detection of Encroachment in Civil Structure Using Dynamic Learning Techniques

Rahul B. Diwate[1], Dr. Lalit V. Patil[2]

[1]Research Scholar, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune, M.H, India. Email: [1]diwate.rahul@gmail.com,

[2]Professor, Department of Information Technology, Smt. Kashibai Navale College of Engineering, Pune, M.H, India Email: [2]lalitvpatil@gmail.com

Article History:

Received: *13-03-2024*

Revised: *17-05-2024*

Accepted: *01-06-2024*

Abstract

Identifying cracks through visual inspection and automated surveys are both effective methods. While both approaches yield satisfactory distress analysis results, automated crack recognition technology stands out for its speed and cost-effectiveness compared to traditional human visual detection methods. This study introduces an innovative approach, namely "crack encroachment in concrete structures classified using DCNN" which employs a Deep Convolutional Neural Network. To enhance the precision of crack detection, the input image undergoes denoising through ADNLMF (Anisotropic Diffusion Non-Local Mean Filtering), preserving edges, textures, and features. Crack discrimination between crack and non-crack images was achieved using Yolo v3, and a deep convolutional neural network classifier was employed to identify specific crack types based on their widths, utilizing crack width transform. This method not only enhances accuracy but also reduces network complexity and processing time. A comprehensive performance comparison with existing crack identification techniques indicates that our proposed methodology for concrete structure crack recognition produces superior results.

Keywords: Automatic crack recognition, Deep Convolutional Neural Network, Anisotropic Diffusion Non-Local Mean Filtering, Crack Width Transform.

1. Introduction

Concrete cracks serve as common indicators of structural wear and tear and are pivotal in infrastructure maintenance. In developed nations, routine assessments of civil engineering structures are conducted, focusing on cracks' presence, position, and width, crucial for developing maintenance strategies [1-3]. Traditionally, manual visual inspections have been the preferred method for gathering such crack data [4]. However, this method is labor-intensive, costly, time-consuming, and often unreliable due to its dependence on the inspector's expertise [5].

To address these challenges, digital image processing has emerged as a viable solution for crack monitoring. This method involves analyzing surface images of concrete buildings to detect vital fracture details, including the presence, location, and width of cracks [6-8]. Various techniques,

including image binarization, edge detection, and mathematical morphology, are employed for crack identification. Image binarization, for example, converts grayscale pixels into black or white, enhancing crack identification. In the binarized images, dark cracks appear black, while lighter backgrounds appear white, facilitating precise crack detection [9]. Additionally, edge detection methods are used to identify the edges of fracture pixels, aiding in concrete crack identification. Mathematical morphology serves as an additional step to modify fracture shapes, further improving the accuracy of crack detection. Previous studies have summarized these image processing methods for identifying fractures in concrete buildings [10-11].

However, despite the promising outcomes of previous research on image processing for crack identification, complete automation is hindered by a fundamental assumption. This assumption presupposes that provided images exclusively contain genuine cracks. For instance, images of a concrete structure's exterior captured manually using a digital camera or an unmanned aerial vehicle (UAV) for structural maintenance might encompass not only cracks but also non-crack elements such as dark stains, shadows, dust, irregularities, and holes. Distinguishing between these diverse features in image processing proves to be challenging [12].

Moreover, the process of image binarization can mistakenly categorize a dark stain as black, resembling a crack, resulting in false-positive outcomes. Therefore, achieving fully automated crack monitoring necessitates the ability to distinguish genuine cracks from surface images that may contain both actual cracks and crack-like non-cracks. In the realm of civil engineering, machine learning has emerged as a transformative technique [13]. Specifically, supervised learning, a subset of machine learning, can be seamlessly integrated with computer vision to tackle challenges associated with crack recognition.

In this approach, unique characteristics of cracks and non-cracks are discerned from training images, which are subsequently utilized in classification algorithms like support vector machines (SVMs) and random forests. The trained classification model is then applied to new images to identify surface cracks effectively [14]. To differentiate between cracks and non-cracks and construct a classification model, features such as geometric patterns (such as eccentricity and the number of pixels in each pixel group) and statistical attributes of pixel intensities (including mean and standard deviation) have been selected. Despite these methods not requiring user-defined empirical thresholds, distinguishing between crack-like and non-cracks that share similar shapes and colors with actual cracks remains a challenge. Therefore, advanced features from both cracks and non-cracks need to be extracted to create a robust classification model for accurate and efficient classification.

The introduction of deep learning, characterized by multiple interconnected layers, has proven to be a potent technique for crack detection. Concrete surface images, categorized as either cracked or undamaged, were employed to construct a classification model based on the Convolutional Neural Network (CNN). This trained model is subsequently applied to evaluate new concrete surface images in the validation process [15]. While previous research utilizing deep learning has successfully identified cracked areas, the challenging task of distinguishing crack-like non-cracks, a common occurrence in real-world scenarios, has not been thoroughly explored [16]. In concrete surface images, it is crucial to precisely detect and filter out any non-crack objects. This paper introduces a framework for concrete crack identification utilizing deep learning, aiming to address these complexities.

The main contributions of this work are

- A collaborative ADNLMF has been used to denoise the input image while maintaining the edges, textures, as well as features.
- The YOLO V3 object detector was used to detect concrete fractures in real-world images, which is far faster than previous detection algorithms with equivalent performance.
- This work develops a less time-consuming and more accurate crack categorization based on crack width.

The following is the layout of the paper: Section 2 contains a brief review of related works, Section 3 contains an overview of the proposed crack detection technique, Section 4 contains the proposed method's implementation outcome, Section 5 comprises the work's conclusion, and the following section contains this research work's references.

2. Literature Survey

Several research studies have been carried out to detect the crack in civil structure which is surveys below.

In a study conducted by Saleem and colleagues [17], an experimental approach was explored to predict the fractured state of concrete surrounding steel reinforcement using ultrasonic pulse velocity testing. To account for the complex and nonlinear stress distribution at the steel-concrete interface, a multilayer feedforward backpropagation perceptron artificial neural network (ANN) was developed. This ANN was designed to avoid oversimplification assumptions when creating models to anticipate cracking. Specifically, the ANN was employed to forecast fracture width, and sensitivity analysis was conducted on various factors influencing bond degradation. The study achieved a high level of accuracy, indicated by an R^2 value of 0.97, showcasing the precision between predicted and experimental values while emphasizing the significance of the most relevant parameter.

In a different approach, Rajadurai and team [18] utilized AlexNet, a pre-trained Deep Convolutional Neural Network (DCNN), for automated vision-based crack recognition and categorization. The methodology involved three key steps: first, gathering numerous images from an open-source picture dataset and classifying them into two categories (non-crack and crack images); second, developing a DCNN model and applying transfer learning and augmentation techniques; and third, automatically identifying and categorizing images using the trained deep learning approach. Furthermore, a cross-dataset study was conducted to evaluate the effectiveness of the trained AlexNet model. The performance of the trained AlexNet model was assessed using precision, recall, accuracy, and F1 measures, demonstrating its effectiveness in crack recognition and categorization.

Hadinata and colleagues [19] conducted a study to assess the effectiveness of advanced encoder-decoder structures in identifying concrete surface fractures using U-Net and DeepLabV3+ architectures, known for their capabilities in biomedical and sparse multiscale image classification, respectively. Cloud computing technology was utilized to train neural networks on a high-performance Graphics Processing Unit NVIDIA Tesla V100 with 27.4 GB of RAM. The study incorporated both internal and external data. Internal data, consisting of basic cracks, was employed for training and validation purposes, while more complex fractures from external sources were used for additional

testing. Both U-Net and DeepLabV3+ architectures were evaluated using four key metrics: accuracy, F1 score, precision, and recall.

In a different study, Ali and team [20] proposed a modified CNN for detecting fractures in concrete structures. The algorithm was compared with four distinct deep learning approaches based on factors such as training data size, data diversity, system complexity, and the number of training epochs. The performance of the proposed convolutional neural network (CNN) model was assessed on eight datasets of varying sizes derived from two public datasets. These results were then compared to outcomes from pre-trained networks, including VGG-16, VGG-19, ResNet-50, and Inception V3 models. Evaluation criteria included computing time, crack localization accuracy, and classification parameters such as accuracy, precision, recall, and F1-score for each model. Notably, on a limited dataset, the customized CNN and VGG-16 models outperformed other methods in terms of classification, localization, and computational efficiency. This indicated that these two models excel in crack detection and localization for concrete structures.

In a study by Chen and colleagues [21], an automated technique for generating crack ground truths (GTs) within concrete images was introduced. The process involves creating initial GTs, pre-training a deep learning-based model, and generating second-round GTs. These second-round GTs are then employed to train a self-supervised crack detection model. Through this method, the pre-trained deep learning-based model demonstrates successful crack detection. A significant contribution of this study is the proposal of an automated GT generation approach, enabling the training of a crack detection model at the pixel level. Experimental results indicate that the second-round GTs closely resemble manually indicated labels.

In another study, Kamada and team [22] developed an Adaptive Structural Deep Belief Network (Adaptive DBN) capable of self-organizing to determine the optimal network structure during the learning process. This hierarchical design incorporates the Adaptive Restricted Boltzmann Machine in each tier (Adaptive RBM). The Adaptive RBM adapts the number of hidden neurons during learning. The proposed technique was applied to the SDNET2018 concrete image dataset, containing approximately 56,000 crack photos from various concrete constructions like bridge decks, walls, and paved roadways. The Adaptive DBN's fine-tuning approach achieved impressive classification accuracies of 99.7%, 99.7%, and 99.4% for three different types of structures. It's worth noting, however, that the database contained some incorrectly labeled data, challenging to assess even for human experts based on photographs.

Chow and colleagues [23] explored the application of deep learning algorithms in civil infrastructure inspection programs. They introduced an AI-powered inspection pipeline that includes anomaly identification, extraction, and fault categorization, replacing the error-prone and time-consuming manual evaluation process. This pipeline generates an anomaly map to extract potential faults, which are then categorized into relevant classes. Utilizing consistent parameters for both anomaly detection and defect categorization in deep learning models ensures unbiased decision-making, eliminating the subjective judgment often associated with human inspectors. Moreover, this approach enhances automation by eliminating the need for multiple time-consuming image processing steps, feature extraction, and selection. The processing time for the image dataset is remarkably quick, approximately 15 minutes.

In a separate study, Shim and team [24] introduced an innovative approach for concrete crack identification based on multiscale and adversarial learning. They constructed a segmentation neural network for precise recognition, incorporating a deep neural network with layer connections to generate additional training data from unlabeled images, thereby enhancing training performance. To minimize the required training data, they devised a novel adversarial learning structure for training multiscale segmentation neural networks, introducing a new loss function to update the weights of these networks.

Additionally, Kumar and colleagues [25] developed a modified LeNet 5 model for identifying fractures in bridges and roads. They employed three distinct datasets: the Automated Bridge Crack Recognition Dataset, the Concrete Crack Imaging for Categorization Dataset, and the Asphalt Crack Dataset. Their proposed approach was evaluated and compared with and without the use of Principal Component Analysis. The model marked crack and non-crack regions in green and red, respectively, considering both time and accuracy elements in the results generation process.

Manual inspection primarily relies on manual inspection, involving the manual depiction of fractures and documentation of their characteristics. However, this method is subjective, dependent on the expertise of the specialist, and lacks objectivity in quantitative analysis. Moreover, existing approaches often have limitations in terms of accuracy, training time, and the incorporation of photographs. In response to these constraints, our research has introduced an innovative model. The following section of this paper offers a detailed overview of our proposed methodology.

3. CRACK CATEGORIZATION TECHNIQUES

The identification and categorization of cracks play a vital role in assessing their severity. Crack detection involves the process of identifying the presence of a crack, while crack classification entails categorizing the crack based on features extracted from the crack region. Therefore, the field of civil infrastructure necessitates automated techniques for both crack detection and classification. Our proposed method, known as "Novel Crack Encroachment in Concrete Structures," is classified using Deep Convolutional Neural Networks (DCNN).

Initially, we preprocess the input image using an innovative approach called Anisotropic Diffusion Non-Local Mean Filtering (ADNLMF). These filters are designed to reduce noise in the input images. Anisotropic Diffusion is employed to maintain the sharpness of edges, while Non-Local Mean denoises the images while preserving textures and other features. The collaborative application of ADNLMF enhances the denoising process, ensuring the preservation of edges, textures, and features.

The preprocessed image is then subjected to a feature extraction step, where we utilize DCNN based on YOLOv3 architecture to detect whether the structure contains cracks or not. YOLOv3 combines elements from YOLOv2, Darknet-53, and Residual networks to extract these features. If a crack is detected, our research further classifies it into categories such as minor crack, major crack, attention crack, or severe crack based on crack depth. The calculation of crack depth is performed using the crack width transform technique. Subsequently, a DCNN-based classifier is utilized to identify the specific type of crack.

In summary, our novel approach enhances accuracy, reduces network complexity, and minimizes processing time in the detection and classification of cracks in civil infrastructure. Figure 1 shows a block diagram of the proposed crack recognition and categorization approach.

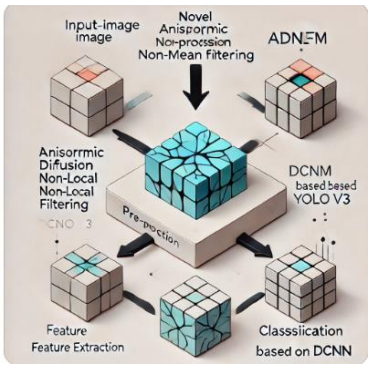

Figure 1: Block diagram of the proposed crack categorization technique

The preprocessing module was given both cracks as well as non-crack pictures to work with. The new Anisotropic diffusion non-local mean filter is used in this module to eliminate noise from pictures. This proposed filtering technique's complete method has been described below.

3.1. Pre-processing

The dataset for implementation is **Concrete Crack Images for Classification** from Mendeley data, which contains 40000 images with 227 x 227 pixels with RGB channels.

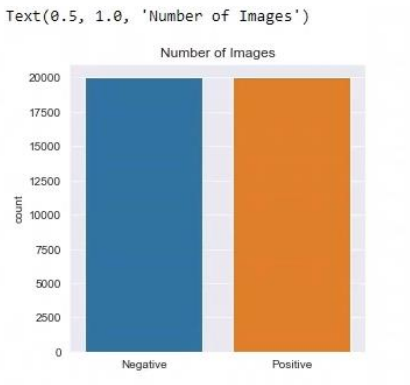

Figure 2. Database of Images

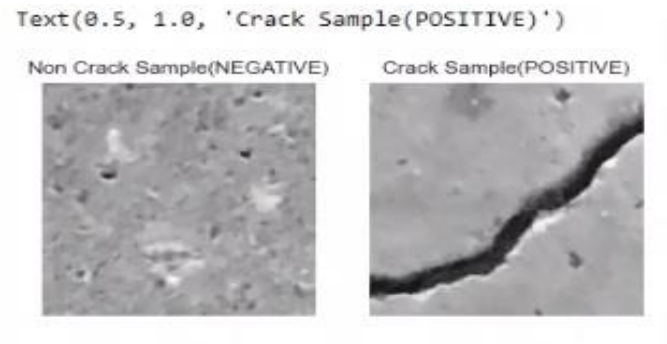

Figure 3. Non-craking and Crack

Images shot fetched from concrete structures are indeed a bit noisy, wherein the concrete structures often suffer from noise, making it necessary to improve the quality of the input images. Various noise-

reduction methods were employed to eliminate unwanted disturbances. While the primary goal of these filters is noise reduction, they must also highlight specific aspects of the input image. Noises in concrete structures typically result from shadows and variations in lighting conditions. Commonly used denoising techniques involve mean filters, median filters, Gaussian filters, and their variations. The Gaussian filter, in particular, is widely used for noise reduction. However, some methods tend to blur the edges of the input images, potentially leading to the loss of crucial information, especially concerning cracked pixels. To address this issue, the proposed approach employs non-linear filtering to remove noise from photographs of concrete buildings, ensuring that the sharpness of the edges is preserved.

3.2. ADNLMF (Anisotropic Diffusion Non-Local Mean Filtering)

The diffusion method is being used to smooth out the input pictures. The diffusion of pixel values influences the approaches and filters offered. Isotropic or anisotropic diffusion processes exist. The diffusion method was applied regardless of any edges in isotropic diffusion. Images get blurry due to the diffusion of pixel values. Isotropic diffusion filters, as a result, average the picture pixels across the entire image. As a result, picture diffusion happens in all directions. Also, the Algorithm for Anisotropic Diffusion Non-Local Mean Filtering has been given below

Algorithm for Anisotropic Diffusion Non-Local Mean Filtering:

Step 1: Input image

Step 2: Give input to the **Anisotropic Diffusion filtering (ADF)**

Step 3: Apply strong Gaussian noise

Step 4: Develop the noisy image.

Step 5: Compute the structural similarity index (SSIM)

Step 6: if SSIM = 1

Step 7: Get the Edge preserving image

Step 8: else

Step 9: Generate **Non-Local Mean Filtering (NLMF)**

Step 10: Add white Gaussian noise with zero mean and variance

Step 11: Develop the noisy RGB image

Step 12: Convert the noisy RGB image to the L*a*b color space

Step 13: Extract a homogeneous L*a*b patch from the noisy background

Step 14: Compute the noise standard deviation

Step 15: Compute the Euclidean distance (e_{dist}) from the origin

Step 16: Calculate the standard deviation of e_{dist} to estimate the noise

Step 17: Degree of Smoothing value to be higher than the standard deviation of the patch

Step 18: Filter the noisy L*a*b* image using NLMF

Step 19: Convert the filtered L*a*b image to the RGB color space.

Step 20: Display the filtered RGB image.

The edge and other pixels are messed together in this scenario. Diffusion varies with direction in anisotropic filters. It's achieved by using an image's gradient. Anisotropic diffusion takes the following general form:

$$\frac{\partial(M(x,y,z))}{\partial(z)} = div\left[s\left(\|\nabla M(x,y,z)\|\middle|\nabla\left(M(x,y,z)\right)\right)\right] \qquad (1)$$

where $M(x,y,z)$ represents the original picture, whereas $\nabla M(x,y,z)$ indicates the gradient of input data at time z.

In Equation (1), $s\ (.)$ seems to be the conductance function that regulates the gradient operation (1). It is quite important in the diffusion process. In general, $s\ (.)$ is not negative. It is chosen depending on the two criteria. If $\lim_{s\to 0} s(x) = 1$, then pixel diffusion is greatest in homogeneous areas. The diffusion pixel values are halted across the edges if $\lim_{s\to 0} s(x) = 0$. It's being used to adjust the diffusion speed and also to assist in differentiating the edges of the input picture. Figure 4 depicts the flowchart for the proposed Anisotropic Diffusion Non-Local Mean Filtering.

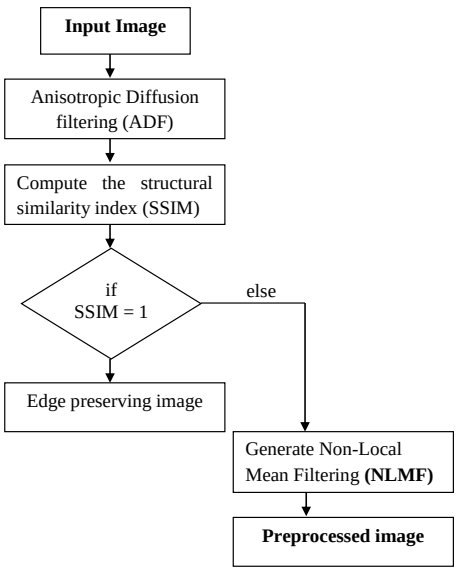

Figure 4: Flowchart for the proposed Anisotropic Diffusion Non-Local Mean Filtering.

The gradient of four $i = \{1, 2, \dots, 4\}$ closest neighbor pixels are utilized to compute the anisotropic diffusion in Equation (1) and also is provided in Equation (2). At iteration z, the anisotropic diffusion filtering of the input may be expressed in terms of diffused image $M_d(x,y,z)$.

$$M_d(x,y,z) = M_d(x,y,z-1) + \frac{1}{4}\sum_{i=1}^{4} m\left(\left|\nabla M_d^i\right|\nabla M_d^i\right), z > 0, \qquad (2)$$

where m represents the diffusion coefficient and ∇ would be the gradient

In general, the Perona-Malik Diffusion (PMD) approach has been widely employed in diffusion. Perona and Malik (1990) presented the diffusion coefficients shown below for anisotropic diffusion filtering:

$$m_1(x) = exp\left[-\left(\frac{x}{n}\right)^2\right] \quad (3)$$

$$m_2(x) = \frac{1}{1+(x/n)^2} \quad (4)$$

The fixed threshold number n governs the degree of input denoising. After every iteration of the PMD technique, the adaptive allocation of the diffusion coefficient $m(x)$ was performed. It isolates the edges in the input picture by lowering the diffusion all along the edge direction. Because the PMD approach employs an unstable diffusion mechanism, artifacts remain in the input image. The threshold n choosing procedure is a difficult assignment. If n is large, the diffusion process generates a smoother picture and blurs the total input image. If n is too little, the diffusion technique yields a smoother image, as well as the abnormalities in the input image, really aren't eliminated. In this case, $M_d(x, y, z)$ is virtually similar to $M_d(x, y, 0)$. The PMD method can successfully smooth parts of a picture having defect-free backdrops. The diffusion coefficients are typically modified in the following steps. Take (x, y) be an image pixel position; at iteration 'i' the grey level probability is determined from the 3×3 neighboring region N_{xy} as follows:

$$J_i^{N_{xy}}(x, y) = \frac{M_i(x,y)}{\overline{M_i}(x,y)} \quad (5)$$

where $\overline{M_\iota}(x, y)$denotes the total grey level in the 3×3 neighboring region N_{xy}. Equation (5) may be recast as a diffusion function, as seen in the equation below:

$$m\left(M_i(x, y), J_i^{N_{xy}}(x, y)\right) = \frac{1}{1+\left(M_i(x,y), J_i^{N_{xy}}(x,y)/n\right)^2} \quad (6)$$

where n is indeed a positive threshold employed as a catalyst for the denoising operation and also to boost the strength of edges. By using the updated diffusion function, Equation (2) may be expressed as Equation (7):

$$M_{d+1}(x, y) = M_d(x, y) + \frac{1}{4}\sum_{i=1}^{4}\left[m\left(M_i(x, y), J_i^{xy}(x, y)\right).M_i(x, y)\right] \quad (7)$$

The improved diffusion function in Equation (6) assists in maintaining the fine features as well as edges of such smoothing operation, preserving the problematic areas. If both J_i^{xy}as well as $M_i(x, y)$ become too big, the diffusion coefficient approaches zero. As a result, the diffusion operation is halted and the associated pixel values were recorded. The picture affects the parameters n but also *i*.

Cracks can be seen in concrete photos against chaotic backgrounds. Cracks of varying diameters and modest slopes are common. The lack of gradients in noisy background photos will have an impact on the accuracy of crack feature recognition. The fractures are kept without damage during the smoothing process using this procedure. $J_i^{xy}(x, y)$ is being used to increase the gradient value while also denoising the input.

The non-local means filtering has been given the picture output from the anisotropic filtering. Non-local means (NLM) denoising seems to be a method for dealing with Gaussian white noise in natural photographs. The essential principle of NLM is to generate the weighting of mean value by assessing the similarities of picture patch pixels, that differ from existing approaches that use single-pixel similarity. As a result, picture denoising with patch information can better preserve image borders, textures, and other properties. Assume there is a noisy image defined by $v = \{v(a)|a \in A\}$, where A specifies the image's coordinate domain. The estimated value of any pixel 'a' in the picture may be determined using NLM by:

$$NLM[v](i) = \sum_{b \in A} w\, f(a,b) v(b) \quad (8)$$

Where the weighting function $\mathrm{wf(a,b)}$ primarily related to the degree of similarity between pixels a, b and fulfills the requirements $0 \leq \mathrm{wf(a,b)} \leq 1$ and $\sum_{\mathrm{b}} \mathrm{wf(a,b)} = 1$

The grey matrices $\mathrm{N_a}$ as well as $\mathrm{N_b}$, which describe the picture areas centered on pixels a and b, correspondingly, define the degree of similarity between pixels a and b. The Gaussian weighted Euclidean distance $dg(a,b)$, may be used to measure the correlation between two areas $\mathrm{N_a}$ and $\mathrm{N_b}$, which is shown as

$$\mathrm{dg(a,b)} = \|\mathrm{v(N_a)} - \mathrm{v(N_b)}\|^2_{2,\vartheta} \quad (9)$$

where ϑ represents the Gaussian kernel's standard deviation. The weighting of comparable pixels in the weighted average increases as the grey matrices of neighboring areas become increasingly similar. The following is the definition of the weighting function $wf(a,b)$:

$$wf(a,b) = \frac{1}{\mathrm{Z(a)}} \mathrm{e}^{-\frac{\mathrm{dg(a,b)}}{\mathrm{r^2}}}$$

$$\mathrm{Z(a)} = \sum_{\mathrm{b}} \mathrm{e}^{-\frac{\mathrm{dg(a,b)}}{\mathrm{r^2}}}$$

where Z(a) is a normalized parameter and r specifies the smoothing parameters, which are connected to the picture noise standard deviation. This collaborative ADNLMF denoises the input picture to appropriately maintain edges, textures, and features, which aids in feature extraction. Figures 5A to 5C display a series of images representing the testing input image, the grayscale version, and the filtered image derived from the images of the civil structure. This set of features comprises training dataset features, testing dataset features, and a validation dataset for these images. The pre-processing of the image involves techniques such as de-noising, sharpening, normalizing, cropping, cleaning, transformation, reduction, quality assessment, and various combinations thereof.

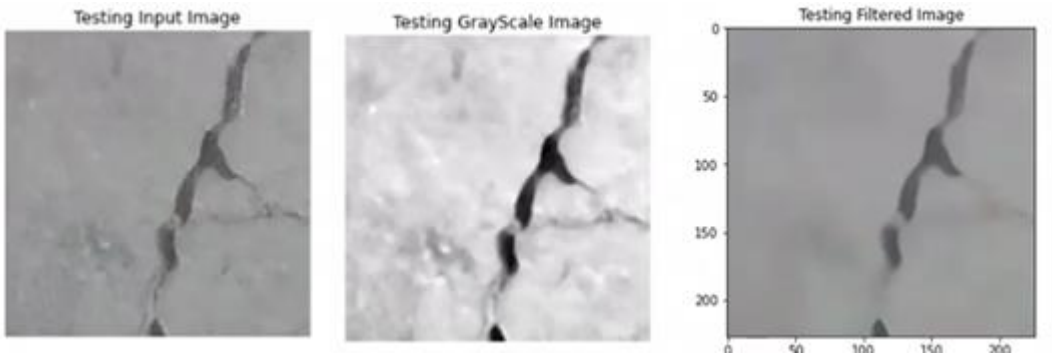

Figure 5A - Figure 5C illustrates a testing input image, testing grayscale image, and a testing filtered image of the images of the civil structure.

The Figure 6A and Figure 6B illustrate a processed image including the input image, and filtered image. The enhanced pre-processed image by means of machine learning, artificial intelligence, deep learning, convolution neural network, deep convolution neural network, and any combination thereof

Figure 6A and Figure 6B illustrate a processed image including the input image, and filtered image.

3.3. Feature extraction

The preprocessed image output is passed through the feature extraction process. In this study, YOLO v3 was employed to extract features, determining whether the image contains cracks or is crack-free.

YOLO v3 represents an enhancement of the YOLO target identification system. This detection approach integrates candidate feature extraction, target categorization, and target localization within a neural network. YOLOv3 treats object identification as a regression problem, predicting class probabilities and bounding box offsets using a single feed-forward convolutional neural network on entire images. Notably, it eliminates area proposal generation and feature resampling, integrating all steps into a single network to establish an end-to-end detection approach. The feature extraction algorithm for YOLO V3 is outlined below.

YOLO-v3 Feature Extraction Algorithm:

Start

Step 1: input image (Output from the preprocessing stage)

Step 2: Ignore the confidence threshold

Step 3: Remaining boxes are undergoing non-maximum suppression

Step 4: Generate default input width and height

Step 5: Read the input image

Step 6: Generate bounding boxes

Step 7: Bounding box confidence level score ≥ confidence threshold

Step 8: Boxes are filtered by Non-Maximum suppression

Step 9: Assign class label and confidence score

Step 10: if a bounding box is generated

Step 11: Extract the crack features

Step 12: else

Step 13: Non-crack

Stop

The input image is divided into $S \times S$ tiny grid cells via the YOLOv3 algorithm. When the center of an item falls within a grid cell, the grid cell is in charge of identifying the object. Every grid cell forecasts the location information of B bounding boxes and then computes the objectness scores associated with these bounding boxes. Each objectivity score could be calculated as follows:

$$C_i^j = P_{i,j}(Object) * IOU_{pred}^{truth} \quad (10)$$

where C_i^j seems to be the objectness score of the jth bounding box inside this ith grid cell. $P_{i,j}(Object)$ is just a function of the object. The IOU_{pred}^{truth} depicts the intersection over union (IOU) between the predicted box as well as the ground truth box. As one component of the loss function, the YOLOv3 technique employs binary cross-entropy of anticipated objectness scores as well as true objectness scores. This can be represented as follows:

$$E_1 = \sum_{i=0}^{S^2} \sum_{j=0}^{B} W_{ij}^{obj} \left[\hat{C}_i^j \, log\left(C_i^j\right) - \left(1 - \hat{C}_i^j\right) log\left(1 - C_i^j\right)\right] \quad (11)$$

S^2 denotes the count of grid cells in the picture, whereas B denotes the count of bounding boxes. The projected abjectness score, as well as the truth abjectness score, were represented by C_i^j and $\hat{C}_i^j$, accordingly. Each bounding box's location was predicated on four forecasts: t_x, t_y, t_w, t_h, with the premise that (c_x, c_y) is indeed the grid cell's offset from the top left corner of the picture. The center point of the final predicted bounding boxes being displaced from the image's top-left corner by (b_x, b_y). These are calculated in the following manner:

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma\left(t_y\right) + y \quad (12)$$

In this case, () is indeed a sigmoid function. The estimated bounding box's width and height were determined as follows:

$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h} \quad (13)$$

where p_w, p_h are indeed the preceding width as well as the height of the enclosing box Dimensional clustering is used to obtain them.

This same ground truth box is typically made up of four parameters: $g_x, g_y, g_w$and g_h, which correspond to the anticipated parameters $b_x, b_y, b_w and\ b_h$. Depending on (12) as well as (13), the truth values of $\hat{t}_x, \hat{t}_y, \hat{t}_w$and $\hat{t}_h$ are as follows:

$$\sigma(\hat{t}_x) = g_x - c_x$$
$$\sigma(\hat{t}_y) = g_y - c_y$$
$$\hat{t}_w = log(g_w/p_w) \quad (14)$$
$$\hat{t}_h = log(g_h/p_h)$$

The square error of coordinate forecasting was being used as one component of the loss function in the Yolo v3 approach. It may be stated as follows:

$$E_2 = \sum_{i=0}^{S^2} \sum_{j=0}^{B} W_{ij}^{obj} \left[\left(\sigma(t_x)_i^j - \sigma(\hat{t}_x)_i^j\right)^2 + \left(\sigma(t_y)_i^j - \sigma(\hat{t}_y)_i^j\right)^2 \right] +$$
$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} W_{ij}^{obj} \left[\left((t_w)_i^j - (\hat{t}_w)_i^j\right)^2 + \left((t_h)_i^j - (\hat{t}_h)_i^j\right)^2 \right] \quad (15)$$

Yolo v3's fundamental categorization network is darknet-53. It makes use of yolov2, Darknet-19, as well as ResNet. This network structure incorporates numerous well-structured 33 and 11 convolution layers, with added shortcut connections in subsequent layers. Consequently, it exhibits outstanding classification performance on ImageNet. Notably, darknet-53 not only delivers similar classification accuracy to ResNet-152 and ResNet-101 but also boasts significantly faster computation speed and fewer network layers. YOLO v3 is a fully convolutional network employing a substantial number of residual layer connections, ensuring the network topology's convergence in deep scenarios and facilitating effective training.

The depth of a network corresponds to the granularity of its expression features and impacts categorization and identification accuracy. Simultaneously, the 1*1 convolution within the residual structure significantly reduces the channel count for each convolution, decreasing the number of features and computational load.

In the context of crack detection, images without cracks lack bounding boxes and need to be segregated, while images containing cracks must be labeled. The classification of crack depth is determined using the Crack Width Transform technique, which is elaborated upon in the subsequent description.

Figures 7A to 7C depict a modified image showcasing the input image, the filtered version, and the identified crack. Figure 7A and Figure 7B are the images considered from Figure 6A and Figure 6 B. This refined pre-processed image is achieved through the utilization of machine learning, artificial intelligence, deep learning, convolutional neural networks, deep convolutional neural networks, and

their various combinations. The processed image is optimized for computational analysis, advantage of the application of these advanced techniques.

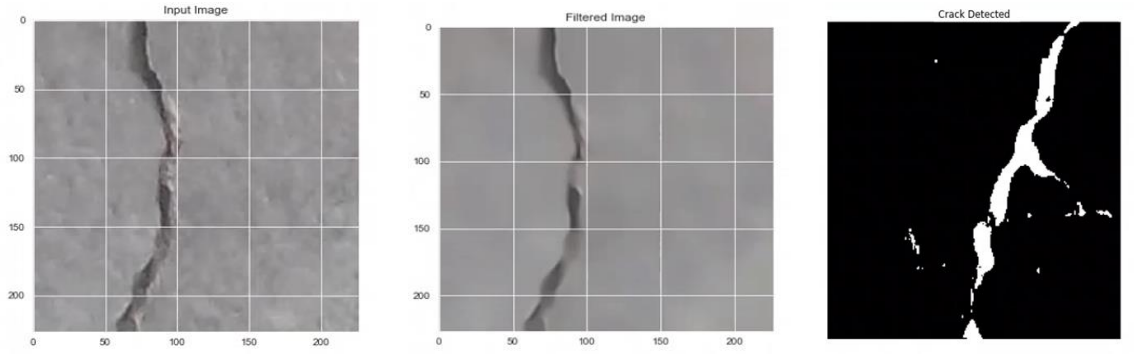

Figure 7A - Figure 7C illustrates a processed image including the input image, filtered image, and crack detected.

3.4. Crack Width Transform

The detected crack images have been sent to the Crack width transform to detect the width of those cracks.

We utilized an edge-based crack width transform(CWT) approach to achieve consistent crack width assessment. The Stroke Width Transform (SWT) is indeed a popular edge-based object extraction approach for character recognition. SWT was composed of three steps: looking for opposing edges, assigning width between opposing edges, and categorizing letters depending on width uniformity. Crack allocation requirements are added to the opposite edge pixel searching method used in the SWT in the CWT. FIGURE 3 depicts the process of looking for an opposite edge pixel (q_j) of any given pixel (q_i) in the normal to the edge direction. As defined by Equation (16), crack width (w) is the number of pixels positioned between the parallel opposing edges.

$$w = \text{card}(q_{ij}) \qquad (16)$$

(q_{ij}) denotes the collection of all pixels across but also (q_i) and (q_j), also $card$ (qij) denotes the count of (qij) (card = cardinality). Iterating the opposite edge pixel lookup till the specified width drops below a maximum threshold value. Considering the restricted crack width permitting measurement, setting the maximum threshold value prevents a non-crack location from being classified as a crack candidate. If the collection of pixels positioned between the opposing pixels (q_{ij}) meets the following characteristics, the extracted opposing edge pixels were classed as potential crack pixels.

1. As defined by Equation (17), the crack width (w) should be less than the maximum threshold value (w_m).

$$w < w_m \qquad (17)$$

2. If the crack width (w) is larger than or equal to the maximum threshold value (w_m), the average Frangi value (fg_{avg}) ought to be higher than that threshold value (fgt), according to Equation (18).

$$fg_{avg} > fg_t \qquad (18)$$

Condition 2 improves the categorization of crack candidate regions by taking the Frangi filtering value into account for crack region enhancement. As described in Equation, the average Frangi filtering value may be determined as the sum of the opposing edge pixels (19).

$$fg_{avg} = \frac{1}{w}\sum_{m=i}^{j} fg(m) \quad (19)$$

The procedure of categorizing the opposing edge pixels as the crack candidate region (C) is expressed in Equation (20).

$$\mathrm{C} = \mathrm{C} \cup \{\mathrm{q_{ij}}\} \quad (20)$$

Narrow-width crack segments have a low intensity, which reduces their Frangi filtering values. To avoid such segments being labeled as non-crack zones, we just applied Condition 1 to narrow-width segments without taking their Frangi filtering values into account. In addition, we created a width map (WM) of the CWT-based crack width measured for aspect ratio filtering. A width map is the alignment of the crack width's midpoints, as stated by Equation (21). When requirements for a candidate crack were met, the width map was initialized as 0, and the crack width was allocated to it.

$$\mathrm{WM}\left(\frac{\mathrm{q_i}+\mathrm{q_j}}{2}\right) \leftarrow \mathrm{w} \quad (21)$$

The extraction of an edge from a crack picture and calculation of its gradient information is referred to as the method of getting crack edge information. To create the crack candidate picture and width map, the CWT is applied to every extracted edge pixel.

3.5. Classification:

Crack classification involves utilizing machine learning algorithms to precisely determine the type of crack. While crack detection identifies the presence of a crack, crack classification categorizes it based on its width. Machine learning, a subset of Artificial Intelligence (AI), enables tasks such as categorization, prediction, and dataset grouping based on specific applications. In this study, Deep Convolutional Neural Network (DCNN) was employed to classify the cracks.

DCNNs replicate the neural connection patterns found in the visual cortex of animals. They comprise at least one convolutional layer, a pooling layer, and a fully connected layer. Each convolutional layer responds to stimuli within a specific part of the visual field, known as its receptive field. This design distinguishes CNNs from traditional image categorization methods and other deep learning techniques, as CNNs can learn filters that are typically hand-crafted in traditional methods. This work utilized 16 convolutional layers and 3 fully connected dense layers. Refer to Figure 8 for the diagram illustrating the DCNN network structure.

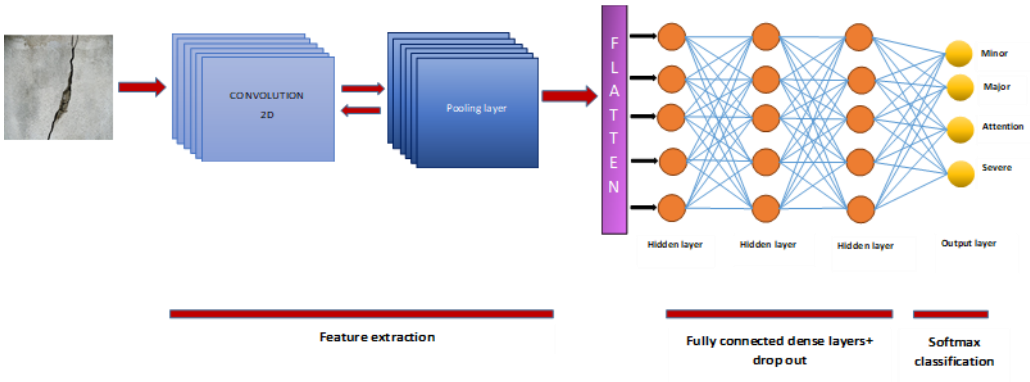

Figure 8: The network model of DCNN

Each convolutional layer was configured with a consistent kernel size of 3×3 pixels, accompanied by padding and a rectified linear unit activation function. Max-pooling layers with strides of 2×2 pixels were employed after the convolutional layers to mitigate position sensitivity issues and enhance the network's general recognition capabilities by extracting feature information from the images. Subsequently, three fully connected hidden layers were established, comprising 1,024, 1,024, and 512 nodes respectively, to capture spatial features and empirically identify the fundamental characteristics of cracks.

A dropout rate of 0.5, a widely used regularization technique for rescaling deep CNN weights to a more effective range, was applied. The final output layer utilized a Softmax classifier to categorize cracks. The training process encompassed 500 epochs, utilizing the Adam algorithm with a learning rate of 0.0001, a stochastic gradient descent approach, to optimize the network weights. After an initial 20 epochs of training, a fine-tuning phase was initiated, adjusting the weights and refining the hyperparameters of the layers to enhance crack classification outcomes.

Advantages:

The proposed system disclosure described herein above has several technical advantages including, but not limited to, the method for detection of encroachment in civil structure using dynamic learning techniques that:

- provides automatic classification of irregularity;
- reduce human intervention;
- reduce error possibility;
- provide high accuracy;
- prevent hazardous conditions occurring by irregularity; and
- provide high-accuracy detection of irregularity.

Conclusion

Surface crack detection is indeed a crucial task in monitoring the structural health of concrete constructions. If cracks form and spread, they limit the effective load-bearing surface area and, in time, can lead to structural failure. The manual crack identification procedure is time-consuming and sensitive to inspectors' subjective opinions. The manual inspection might also be challenging in the case of high-rise structures and bridges. This paper proposes an automated crack identification approach called crack encroachment in concrete structures identified with DCNN (Deep Convolutional Neural Network). It denoises the input image using ADNLMF (Anisotropic Diffusion Non-Local Mean Filtering) to better preserve the edges, textures, and features. Yolo v3 was used to discriminate between crack and non-crack images, and a deep convolutional neural network classifier was used to detect the crack type. The crack kinds are recognized here based on the width of the cracks by utilizing crack width transform. This method not only improves accuracy but also decreases network complexity as well as time. When the overall performance of our proposed method is compared to that of existing crack identification techniques, it is clear that our proposed concrete structure crack recognition methodology produces better results.

References

[1] Haynes C, Todd MD, Flynn E, et al. Statistically-based damage detection in geometrically-complex structures using ultrasonic interrogation. Struct Health Monit 2013; 12(2): 141–152.

[2] Larrosa C, Lonkar K and Chang FK. In situ damage classification for composite laminates using Gaussian discriminant analysis. Struct Health Monit 2014; 13(2): 190–204.

[3] Qiu L, Yuan S and Boller C. An adaptive guided wave Russian mixture model for damage monitoring under time-varying conditions: validation in a full-scale aircraft fatigue test. Struct Health Monit 2017; 16(5): 501–517.

[4] Liu P, Lim HJ, Yang S, et al. Development of a ‘‘stickand-detect’’ wireless sensor node for fatigue crack detection. Struct Health Monit 2017; 16(2): 153–163.

[5] Karthick SP, Muralidharan S, Saraswathy V, et al. Effect of different alkali salt additions on concrete durability property. J Struct Integ Maint 2016; 1(1): 35–42.

[6] Domaneschi M, Sigurdardottir D and Glisˇic´ B. Damage detection based on output-only monitoring of dynamic curvature in concrete-steel composite bridge decks. Struct Monit Maint 2017; 4(1): 1–15.

[7] Xu J, Fu Z, Han Q, et al. Micro-cracking monitoring and fracture evaluation for crumb rubber concrete based on acoustic emission techniques. Struct Health Monit. Epub ahead of print 15 September 2017. DOI: 10.1177/1475921717730538.

[8] Reagan D, Sabato A and Niezrecki C. Feasibility of using digital image correlation for unmanned aerial vehicle structural health monitoring of bridges. Struct Health Monit. Epub ahead of print 10 October 2017. DOI: 10.1177/1475921717735326.

[9] Hu WH, Said S, Rohrmann RG, et al. Continuous dynamic monitoring of a prestressed concrete bridge based on strain, inclination, and crack measurements over 14 years. Struct Health Monit. Epub ahead of print 30 October 2017. DOI: 10.1177/1475921717735505.

[10] Liu Y, Cho S, Spencer BF Jr, et al. Automated assessment of cracks on concrete surfaces using adaptive digital image processing. Smart Struct Syst 2014; 14(4): 719–741.

[11] Kim H, Ahn E, Cho S, et al. Comparative analysis of image binarization methods for crack identification in concrete structures. Cement Concrete Res 2017; 99: 53–61.

[12] Kim H, Lee J, Ahn E, et al. Concrete crack identification using a UAV incorporating hybrid image processing. Sensors 2017; 17(9): E2052.

[13] Abdel-Qader I, Abudayyeh O and Kelly ME. Analysis of edge-detection techniques for crack identification in bridges. J Comput Civil Eng 2003; 17(4): 255–263.

[14] Hutchinson TC and Chen Z. Improved image analysis for evaluating concrete damage. J Comput Civil Eng 2006; 20(3): 210–216.

[15] Jahanshahi MR, Masri SF, Padgett CW, et al. An innovative methodology for detection and quantification of cracks through the incorporation of depth perception. Mach Vision Appl 2013; 24(2): 227–241.

[16] Lee BY, Kim YY, Yi S-T, et al. Automated image processing technique for detecting and analyzing concrete surface cracks. Struct Infrastruct Eng 2013; 9(6): 567–577.

[17] Saleem, Muhammad, and Hector Gutierrez. "Using an artificial neural network and non-destructive test for crack detection in the concrete surrounding the embedded steel reinforcement." Structural Concrete 22, no. 5 (2021): 2849-2867.

[18] Rajadurai, Rajagopalan-Sam, and Su-Tae Kang. "Automated Vision-Based Crack Detection on Concrete Surfaces Using Deep Learning." Applied Sciences 11, no. 11 (2021): 5229.

[19] Hadinata, Patrick Nicholas, Djoni Simanta, Liyanto Eddy, and Kohei Nagai. "Crack Detection on Concrete Surfaces Using Deep Encoder-Decoder Convolutional Neural Network: A Comparison Study Between U-Net and DeepLabV3+." In Journal of the Civil Engineering Forum, vol. 7, no. 3, pp. 323-334. 2021.

[20] Ali, Luqman, Fady Alnajjar, Hamad Al Jassmi, Munkhjargal Gochoo, Wasif Khan, and M. Adel Serhani. "Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures." Sensors 21, no. 5 (2021): 1688.

[21] Chen, Hsiang-Chieh, and Zheng-Ting Li. "Automated Ground Truth Generation for Learning-Based Crack Detection on Concrete Surfaces." Applied Sciences 11, no. 22 (2021): 10966.

[22] Kamada, S., Ichimura, T. and Iwasaki, T., 2020, March. An Adaptive Structural Learning of Deep Belief Network for Image-based Crack Detection in Concrete Structures Using SDNET2018. In *2020 International Conference on Image Processing and Robotics (ICIP)* (pp. 1-6). IEEE.

[23] Chow, J.K., Su, Z., Wu, J., Li, Z., Tan, P.S., Liu, K.F., Mao, X. and Wang, Y.H., 2020. Artificial intelligence-empowered pipeline for image-based inspection of concrete structures. *Automation in Construction*, *120*, p.103372.

[24] Shim, S., Kim, J., Cho, G.C. and Lee, S.W., 2020. Multiscale and adversarial learning-based semi-supervised semantic segmentation approach for crack detection in concrete structures. *IEEE Access*, *8*, pp.170939-170950.

[25] Kumar, A., Kumar, A., Jha, A.K. and Trivedi, A., 2020, December. Crack detection of structures using a deep learning framework. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)* (pp. 526-533). IEEE.