infrastructures

MDPI

Article

Computer Vision-Based Concrete Crack Identification Using MobileNetV2 Neural Network and Adaptive Thresholding

Li Hui [1], Ahmed Ibrahim [2,*] and Riyadh Hindi [3]

[1] Civil Engineering Department, University of Louisiana at Lafayette, Lafayette, LA 70504, USA; li.hui@louisiana.edu
[2] Civil & Environmental Engineering Department, University of Idaho, Moscow, ID 83844, USA
[3] Department of Civil, Computer and Electrical Engineering, Saint Louis University, St. Louis, MO 63103, USA; riyadh.hindi@slu.edu
* Correspondence: aibrahim@uidaho.edu

Abstract: Concrete is widely used in different types of buildings and bridges; however, one of the major issues for concrete structures is crack formation and propagation during its service life. These cracks can potentially introduce harmful agents into concrete, resulting in a reduction in the overall lifespan of concrete structures. Traditional methods for crack detection primarily hinge on manual visual inspection, which relies on the experience and expertise of inspectors using tools such as magnifying glasses and microscopes. To address this issue, computer vision is one of the most innovative solutions for concrete cracking evaluation, and its application has been an area of research interest in the past few years. This study focuses on the utilization of the lightweight MobileNetV2 neural network for concrete crack detection. A dataset including 40,000 images was adopted and preprocessed using various thresholding techniques, of which adaptive thresholding was selected for developing the crack evaluation algorithm. While both the convolutional neural network (CNN) and MobileNetV2 indicated comparable accuracy levels in crack detection, the MobileNetV2 model's significantly smaller size makes it a more efficient selection for crack detection using mobile devices. In addition, an advanced algorithm was developed to detect cracks and evaluate crack widths in high-resolution images. The effectiveness and reliability of both the selected method and the developed algorithm were subsequently assessed through experimental validation.

Keywords: structural inspection; concrete crack evaluation; computer vision; mobilenetv2; adaptive thresholding

Academic Editors: Pedro Arias-Sánchez and Alessandro Zona

Received: 11 December 2024
Revised: 10 February 2025
Accepted: 11 February 2025
Published: 18 February 2025

Citation: Hui, L.; Ibrahim, A.; Hindi, R. Computer Vision-Based Concrete Crack Identification Using MobileNetV2 Neural Network and Adaptive Thresholding. *Infrastructures* **2025**, *10*, 42. https://doi.org/10.3390/infrastructures10020042

1. Introduction

Concrete is widely selected for diverse types of infrastructures, ranging from monolithic skyscrapers to complex bridge systems due to its notable compressive strength, environmental resilience, and adaptability in form. However, one of the major issues for concrete structures is the cracks during its service life. Concrete cracking can arise from a myriad of reasons, from physiochemical processes such as hydration-induced shrinkage to external factors like mechanical stresses and environmental conditions, leading to significant challenges for structural inspection and health monitoring [1,2]. Those cracks can potentially introduce harmful agents like chlorides, carbonates, and moisture into the structures, affecting the structural integrity and reducing the overall lifespan of the concrete members [3].

Traditional methods for crack detection primarily depend on manual visual inspection, which relies on the experience and expertise of inspectors using tools such as magnifying glasses, microscopes, and calipers to identify surface defects and crack lengths and widths. While this hands-on approach has been a trusted method for years, it is often time-consuming, labor-intensive, and can be subject to human error or oversight, especially in areas that are difficult to access or have complex geometries [4–8]. To address these issues, researchers have explored non-destructive testing techniques like ultrasonic pulse velocity, impact echo, and ground-penetrating radar, which offer better accuracy and can cover larger areas more efficiently [9–16]. However, these techniques require specialized equipment and trained personnel, which can significantly increase the overall inspection cost.

In addressing the complexities of infrastructure evaluation, computer vision, a specialized branch of artificial intelligence dedicated to equipping machines with visual data processing capabilities, has emerged as a significant innovation [17]. Integrating infrastructure monitoring and inspection with computer vision paves the way for revolutionary methods that surpass the limitations of traditional manual vision inspection. Thus, large-scale inspections can be conducted with a fraction of the manpower and time traditionally requisitioned, offering both efficiency and breadth through the deployment of computer vision systems such as unmanned aerial vehicles (UAVs) [2,8]. In addition, the precision and consistency of computer vision algorithms distinctly differ from human-led inspections. Human structural inspectors, despite their expertise, can sometimes be influenced by subjective biases, variability, and fatigue. In contrast, computer vision algorithms deliver consistent performance, and document defects while providing measurable data that enhance assessment precision [4,18].

The application of computer vision in concrete crack detection has been an area of research interest in the past few years. Several researchers have explored diverse techniques to achieve high accuracy and reliability in detecting defects, especially cracks in concrete structures. Koch et al. (2015) provided an extensive review of computer vision-based methods for automated crack detection on concrete and asphalt surfaces [17]. Their discussion spanned across methods ranging from traditional image processing to more advanced machine learning techniques, highlighting the progression and challenges in the domain. Subsequently, Cha et al. (2017) proposed a crack detection method that employed deep convolutional neural networks (CNNs) [19]. They trained the network using extensive datasets to recognize the patterns inherent to concrete cracks. Their method was significant as it demonstrated the robustness and potential of deep learning techniques for concrete surface inspections. Zhong et al. (2018) explored the application of convolutional neural networks combined with unmanned aerial systems for bridge inspections [20]. Their methodology integrated the scalability and efficiency of drones with the precision of computer vision techniques, facilitating comprehensive and efficient structural assessments. Meanwhile, with the rapid advancements in technology in recent years, there has been an explosion in the development of sophisticated deep-learning algorithms for concrete defect detection using different types of CNN-like neural networks, such as two-stage CNN model, back-propagation (BP) neural network, and you-only-look-once (YOLO) algorithm [18,21–31].

Using deep learning techniques with computer vision, the field of concrete crack detection has seen significant advancements in recent years. For instance, recent works have demonstrated the application of AI-empowered inspection pipelines for robust concrete structures, even under challenging conditions [32]. Moreover, a comprehensive framework for automated concrete structure inspection, integrating mobile data collection with 360° cameras and LiDAR, defect detection using deep learning, scene reconstruction, defect assessment, and Building Information Modeling for streamlined facility management, validated by

a case study on concrete cracks and spalls [33]. As explored in this work, these contributions form a strong foundation for advancing lightweight models like MobileNetV2.

The focus of this study revolves around the utilization of the MobileNetV2 neural network for concrete crack detection. MobileNetV2, which is an evolution of its precursor MobileNetV1, symbolizes the advancements in deep learning models designed specifically for mobile and embedded vision platforms [34]. The major advantage of MobileNetV2 lies in its adeptness in offering an optimal equilibrium between computational economy and performance. Such a balance ensures that the model remains resilient in various vision assignments without sacrificing processing speed or model compactness [35]. However, very few studies have been conducted to evaluate the performance of MobileNetV2 in concrete defect detection, especially cracks.

In this research, the MobileNetV2 neural network was employed as the primary methodology for concrete crack detection. A dataset of 40,000 images was adopted and preprocessed using various thresholding techniques, of which adaptive thresholding was selected for developing the crack evaluation algorithm. While both CNN and MobileNetV2 indicated comparable accuracy levels in crack detection, the MobileNetV2 model's significantly smaller size set it apart as a more efficient selection. In addition, an advanced algorithm was developed to detect cracks and evaluate their width in high-resolution images. The effectiveness and reliability of both the selected method and the developed algorithm were subsequently assessed through comprehensive experimental validation.

2. Dataset and Preprocessing

2.1. Dataset Description

The dataset used in this study is a comprehensive collection developed by Ozgenel [36], which includes 40,000 high-resolution images of concrete surfaces. These images represent a diverse range of scenarios, from uncracked concrete surfaces to those with cracks of varying intensities. Each image in the dataset has a resolution of 227 × 227 pixels, offering a detailed view crucial for concrete crack detection. The images within the dataset are systematically classified into two distinct subsets—"Positive" and "Negative". The "Positive" subset includes images of concrete surfaces with visible cracks, representing the challenges often encountered in real-world concrete structures. Conversely, the "Negative" subset comprises images of uncracked concrete surfaces, serving as a control group to help train the model in recognizing surfaces without defects. A set of sample images is shown in Figure 1.

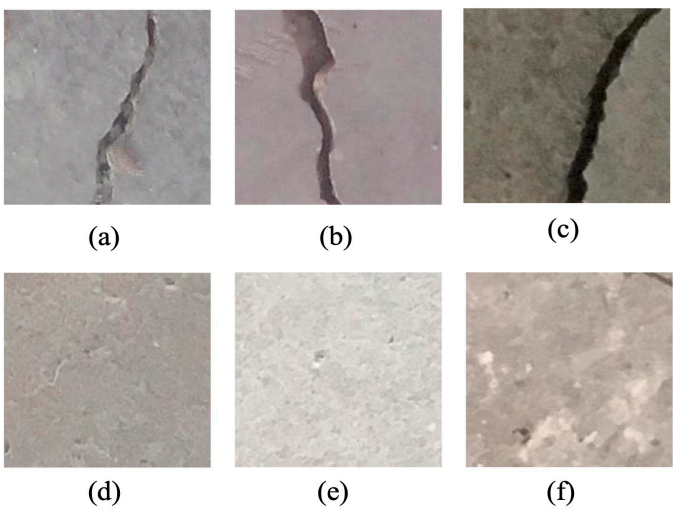


Figure 1. Sample dataset of concrete surfaces with crack (**a**–**c**) and without crack (**d**–**f**).

A notable feature of this dataset is its extensive coverage and variety. With 40,000 images, it ensures that models trained on it are exposed to various scenarios. Furthermore, the consistency in the resolution across images guarantees that the model receives uniform input, removing potential biases or discrepancies that might arise due to varying image sizes. By presenting both "Positive" and "Negative" images in equal numbers and maintaining a consistent resolution, the dataset establishes a balanced and rigorous training environment. The extensive collection of images within the dataset ensures a consistent training process, leading to enhanced model accuracy and generalization capabilities in real-world scenarios.

2.2. Image Preprocessing

Image preprocessing plays an important role in ensuring that machine learning models receive clean and standardized input data, which can considerably enhance their performance. Before the main analysis, images are processed using multiple techniques to optimize their quality and ensure consistency. In the image preprocessing phase, grayscale conversion is implemented followed by advanced thresholding techniques. Grayscale conversion reduces computational complexity by eliminating color channels, enabling the model to concentrate on the inherent structural patterns and contrasts that are crucial for identifying cracks in concrete. Thresholding simplifies image content by segmenting it into distinguishable features, enhancing contrast, and facilitating efficient and consistent computational analysis. Figure 2 shows some concerns in the unprocessed images with surface textures, colors, and illumination issues.

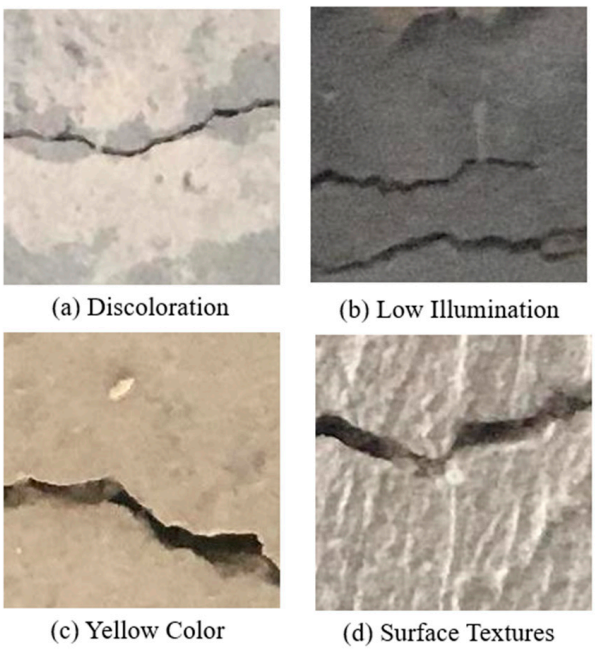


Figure 2. Issues in the dataset.

2.2.1. Grayscale Conversion

One of the primary steps during the image preprocessing is the conversion of images to grayscale. Grayscale conversion is the process of transforming a full-color image into shades of gray, ranging from black to white. In the context of concrete crack detection, the surface textures, shades, and patterns of the concrete are often more critical than the actual color information. By converting images to grayscale, the color data are removed, allowing the model to focus more on the textual and structural features in the image, which can significantly affect the accuracy during crack identification.

The conversion to grayscale was achieved by taking a weighted sum of the three RGB channels (Red, Green, and Blue) for every pixel. The formula used in this study is as follows [37]:

$$Grayscale\ Value = 0.2989 \times Red + 0.5870 \times Green + 0.1140 \times Blue \quad (1)$$

The weights used in the grayscale conversion equation are based on the luminance values that account for the human perception of colors [38]. Human vision does not perceive all colors with equal sensitivity. Specifically, our eyes are more sensitive to green light, followed by red light, and are less sensitive to blue light. These weights ensure that the grayscale image's brightness corresponds closely to the perceived brightness of the original color image, making it more in line with how a human observer would perceive the scene in monochrome. Figure 3 shows the grayscale processing for the dataset used in this study.

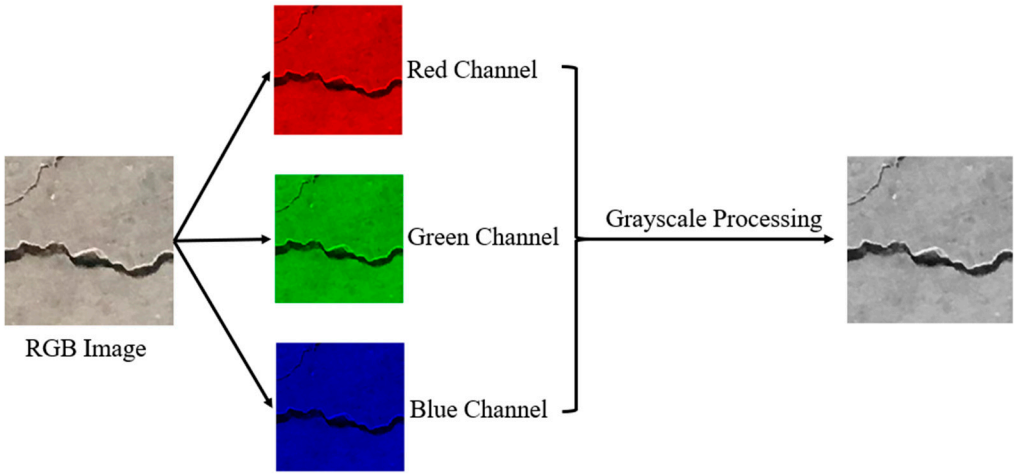


Figure 3. Grayscale Processing for the Dataset.

Grayscale conversion offers several advantages during image processing tasks. First, the conversion emphasizes luminance over chrominance, enabling models to focus on concrete crack detection rather than shade, pattern of concrete, etc. This is particularly advantageous as cracks manifest predominantly as variations in luminance rather than color deviations. Moreover, by converting images to grayscale, computational resources are optimized since grayscale images require fewer computational parameters than their colored counterparts. This efficiency is especially pertinent when processing high-resolution images. Furthermore, the human-perception-based weighting provides a standardized brightness perception, ensuring that the grayscale images closely resemble how a human observer would perceive the scene in monochrome. This is beneficial in applications where the end goal is human interpretation or assessment. Additionally, grayscale images offer consistency across datasets, eliminating potential biases introduced by varying color hues in materials like concrete, which can change color due to age, moisture, or constituent materials.

2.2.2. Thresholding

Thresholding is a technique designed to split images into distinct regions based on pixel intensity values in image processing. Through this method, an image is transformed by assigning each pixel a new value, depending on whether its current value is above or below a specified threshold. The outcome is an enhanced contrast between regions of interest, making them more distinguishable from their surroundings.

During concrete crack detection, concrete surfaces exhibit different surface patterns, textures, and occasional discolorations, all of which can dramatically affect the accuracy of automated analysis techniques. The application of thresholding techniques in this study

highlights cracks by focusing on intensity differences against the uniform background of the concrete. Thresholding not only helps with the identification of cracks but also improves the conditions of the image for deep learning by reducing noise and eliminating redundant features. This optimization directly leads to increased precision in crack identification, highlighting thresholding's critical importance in this research. A total of six thresholding methodologies were assessed and compared in this study, with Table 1 details the specifics of each technique.

Table 1. Comparison of Thresholding Methods [39].

No.	Thresholding Approach	Optimal for	Computation Time	Sensitivity to Noise	Note
1	Global	Uniform Illumination	Fast	High	Simple, fixed threshold for whole image
2	OTSU	Bi-model Histograms	Moderate	Moderate	Maximizes interclass variance
3	Adaptive	Variable Illumination	Slow	Low	Threshold varies over image regions
4	Triangle	Sparse Histograms	Moderate	High	Useful for images with sparse histogram
5	Isodata	Multi-Model Histogram	Moderate	Moderate	Iteratively refines the threshold
6	Gaussian Mixture Model	Complex Histogram	Slow	Low	Use Gaussian distribution to model background and foreground

Six thresholding methods were compared in this study. Global thresholding, best suited for images with uniform illumination, offers a fast but noise-sensitive approach by setting a fixed threshold for the entire image. Otsu's method optimally segments bi-modal histograms by maximizing interclass variance. For images with inconsistent lighting, adaptive thresholding adjusts the threshold across different regions, although at a slower computational pace. Triangle thresholding is tailored for images with sparse histograms, while the Isodata method continually refines the threshold, which makes it ideal for multi-modal histograms. The Gaussian Mixture Model (GMM) employs Gaussian distributions to depict the background and foreground, making it apt for images with intricate histograms. Figure 4 shows the comparison of different thresholding approaches.

In this study, adaptive thresholding was finally chosen as the noise reduction technique during the crack detection process. Utilizing adaptive thresholding brought several advantages. Firstly, unlike global thresholding which uses a single value for the entire image, adaptive thresholding determines the threshold for a pixel based on a small region around it. This ensures that the method can handle variations in lighting or shadow effects more effectively. Additionally, it proves beneficial for images with varying background intensities. Adaptive thresholding's ability to adjust to localized image regions makes it particularly effective in detecting subtle cracks often missed by other methods. Figure 5 shows the results of adaptive thresholding applied with different parameter settings. These parameters need to be carefully calibrated during the crack detection process to ensure precision in various scenarios.

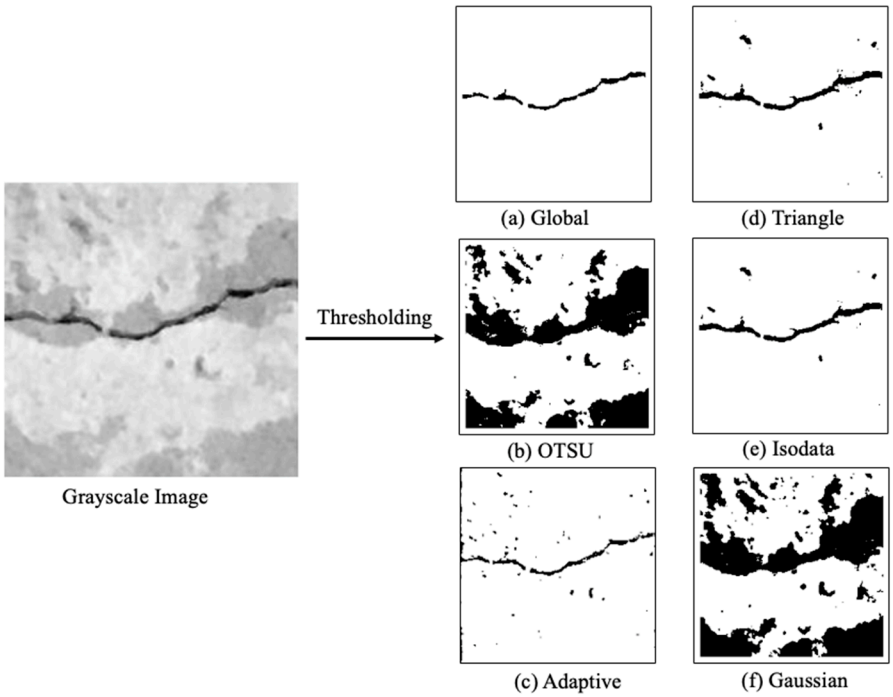


Figure 4. Thresholding using Different Approach: (**a**) Global; (**b**) OTSU; (**c**) Adaptive; (**d**) Triangle; (**e**) Isodata; (**f**) Gaussian.

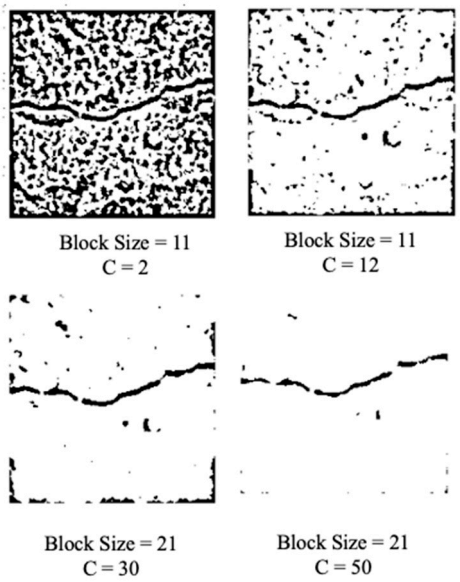


Figure 5. Adaptive Thresholding with Different Block Sizes and Constant C.

2.2.3. Normalization

Prior to feeding the images into the neural network, the total pixel values ranged from 0 to 255 were converted to a scale between 0 and 1. Such normalization ensures faster convergence during training and helps in maintaining consistent weight updates, especially for neural networks like MobileNetV2. In this study, normalization was implemented using the following equation:

$$Normalized\ Pixel\ Value = \frac{Original\ Pixel\ Value - Minimum\ Pixel\ Value}{Maximum\ Pixel\ Value - Minimum\ Pixel\ Value} \quad (2)$$

For the dataset used in this study, the maximum pixel value is 255 and the minimum pixel value is 0. Therefore, the normalization equation can be simplified to the following:

$$Normalized\ Pixel\ Value = \frac{Original\ Pixel\ Value}{255} \quad (3)$$

As shown in Figure 6, all images are brought to a consistent scale, regardless of their original lighting conditions or capture settings. With more consistent data, the neural network can learn features more effectively, leading to better generalization and potentially higher accuracy during validation and testing.

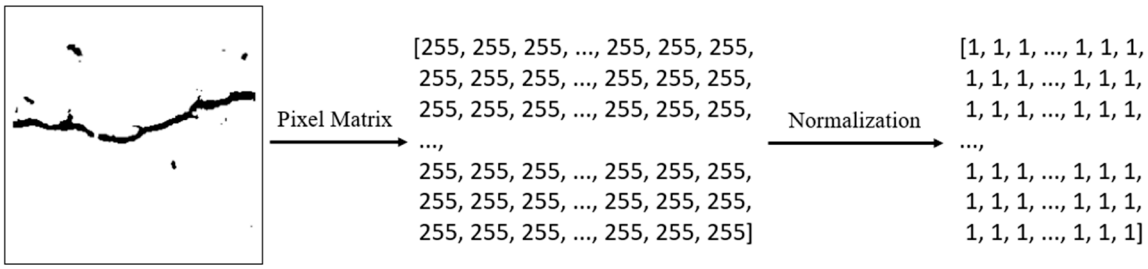


Figure 6. Example of Normalization Process for Image after Thresholding.

3. Development of Deep Learning Model

3.1. Architecture of MobileNetV2 and CNN Neural Networks

The efficiency and effectiveness of the MobileNetV2 architecture were optimized to address the task of concrete crack identification in this study. The architecture's unique design principles, optimized for mobile and embedded devices, proved beneficial for processing high-resolution concrete images, which often present challenges due to their intricate patterns and textures. The architecture of the MobileNetV2 neural network used in this study is shown in Figure 7.

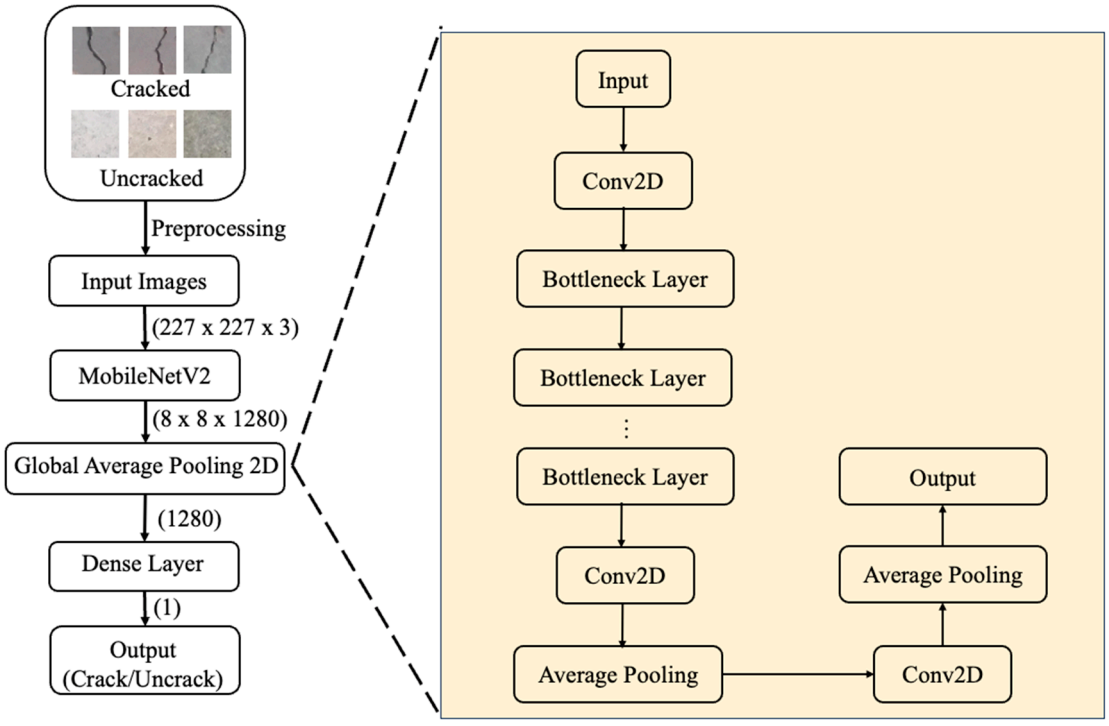


Figure 7. MobileNetV2 Neural Network Architecture.

The MobileNetV2 model developed in this study accepts input images of size $227 \times 227 \times 3$ pixels. The images were preprocessed by conducting grayscale, thresholding, and normalization to ensure the optimal format for training. The initial layer of the

MobileNetV2 block consists of a 3 × 3 convolutional layer equipped with 32 filters. This initial layer is vital for capturing the primary features of the input image and sets the stage for the subsequent bottleneck layers.

One of MobileNetV2's signature features is its bottleneck residual blocks. In this study, several bottleneck residual blocks were utilized in sequence. Each block typically undergoes an expansion of channels using 1 × 1 convolutions, followed by depth-wise convolutions and a projection back to the original channel depth. This strategy, termed as inverted residuals, drastically reduces computational cost without compromising feature extraction capability. The ReLU activations are omitted since each bottleneck block culminates in a linear bottleneck. This ensures that small-scale features crucial for crack identification are preserved. However, ReLU was employed in other layers to further enhance the architecture's efficiency, which makes it particularly well suited for the low-precision computations that our study aimed for.

A global average pooling layer was used to minimize overfitting and ensure spatial invariance. This was followed by a fully connected layer with a sigmoid activation function to better fit our binary classification task, which was used to identify the presence or absence of cracks. Inherent to the MobileNetV2 architecture, the neural network model developed in this study resulted in a compact size model which required significantly fewer computations compared to larger architectures. This not only accelerated our training process but also ensured that the model could be deployed in real-world, resource-constrained environments for on-site concrete inspections.

The performance metrics derived from the MobileNetV2 model were compared with those achieved using a traditional convolutional neural network (CNN). This comparative analysis was to highlight the relative performance and efficiency of the two architectures. Moreover, an important objective of this study was the potential deployment of the neural network model on mobile devices. To this end, the physical size of the models would also come into consideration, as mobile deployment necessitates compact and lightweight models for seamless integration. The structure of the CNN model is shown in Figure 8.

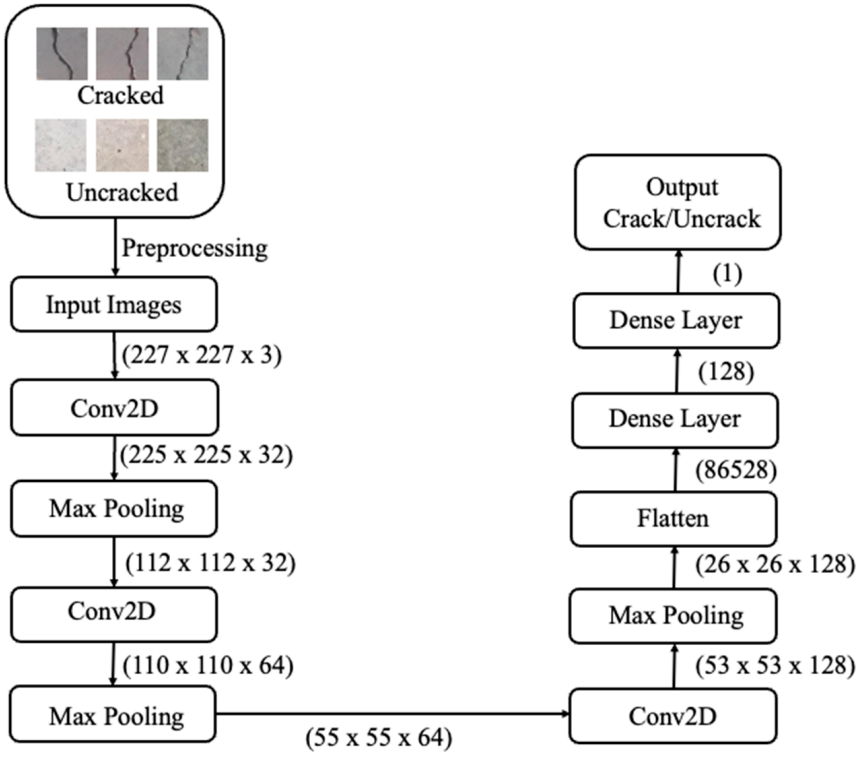


Figure 8. CNN Neural Network.

3.2. Results and Comparison

The dataset was divided into two subsets, namely training and validation. Up to 80% (32,000) of the images were allocated to the training set and 20% (8000) of the images were assigned to the validation set. Partitioning is essential to prevent overfitting and to ensure the model's performance metrics. An in-depth comparison was conducted to evaluate the efficiency of various thresholding methods and the performance of CNN model in concrete crack detection. The comparisons of training and validation accuracies are shown in Figures 9 and 10. It was observed that most thresholding techniques yielded remarkably similar results. After training for 10 epochs, these methods consistently achieved an accuracy around 99%. The method using grayscale images without any thresholding provides an accuracy of 96%, which demonstrates the effectiveness of thresholding in refining image quality and enhancing detection precision.

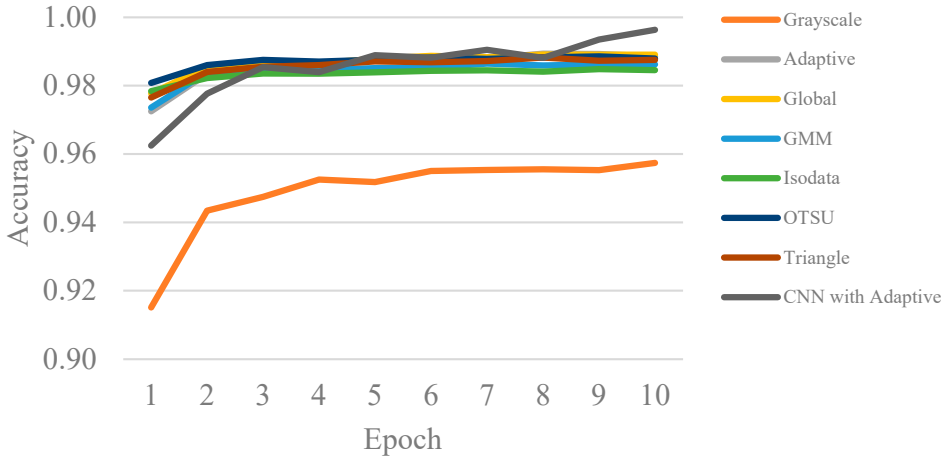


Figure 9. Comparison of Training Accuracy.

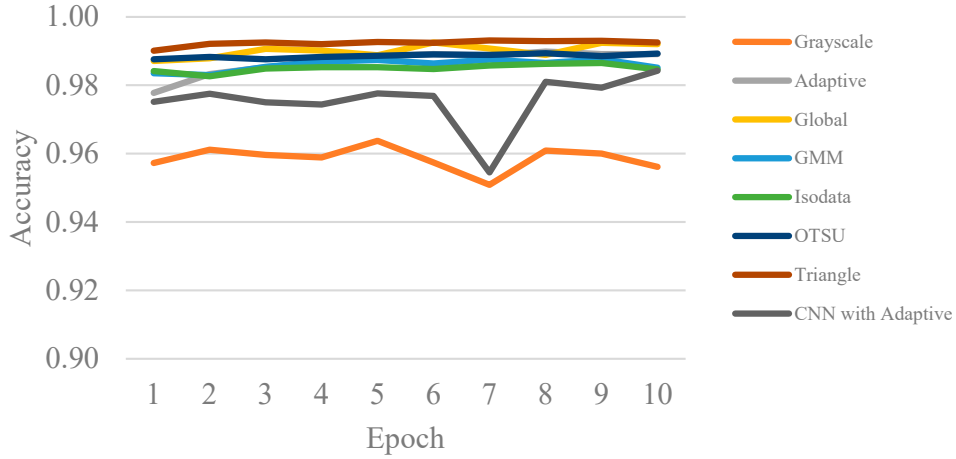


Figure 10. Comparison of Validation Accuracy.

The traditional CNN model was also compared with the MobileNetV2 model developed in this study. The accuracy metrics of the CNN model were closely aligned with those obtained through MobileNetV2 models with thresholding. However, a significant difference was noted in the comparison of model size. The CNN model, despite its high accuracy, had a substantial model size of 134.1 MB. In contrast, the MobileNetV2 model, which delivered comparable accuracy, was significantly more compact with a size of just 9.4 MB.

4. Development of Crack Identification Algorithm

4.1. Overview of Crack Identification Algorithm

During structural inspection or health monitoring, high-resolution images are commonly used in order to obtain fine details, especially when identifying minor cracks or defects. However, these high-resolution images bring issues such as computational burdens and model compatibility issues. To leverage the detailed nature of these images and the specific input dimensions expected by previously developed MobileNetV2 frameworks, the sliding window technique was adopted in this study as a potential solution. The sliding windows technique, by systematically parsing images into overlapping segments, ensures comprehensive crack detection while aligning with model input requisites.

However, the utilization of the sliding window technique introduces the potential redundancy of detecting the same crack multiple times due to window overlaps. The Non-Maximum Suppression (NMS) method was adopted to address this issue through crack localization methodology. This approach consolidates redundant detections, offering a precise representation of each crack. Furthermore, the algorithm tries to evaluate the width of each crack, translating the crack's width in pixel to real-world dimensions by capitalizing on the known camera–concrete distance. Figure 11 shows the basic procedures for crack identification.

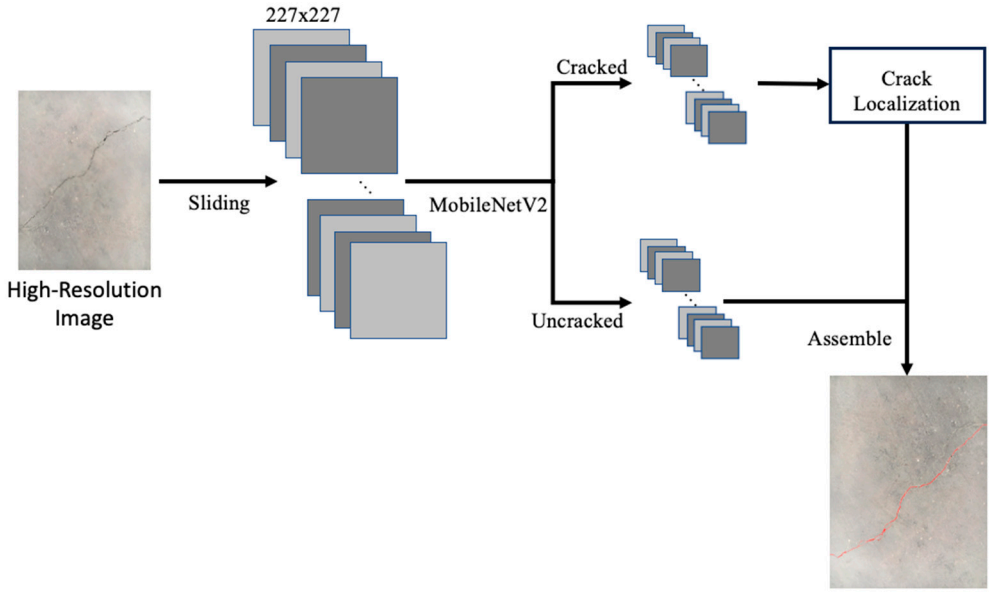


Figure 11. Framework for Crack Identification.

4.2. Sliding Windows and Non-Maximum Suppression

As shown in Figure 12, the sliding (movable) windows technique was adopted in developing the crack identification algorithm to dissect the high-resolution image into smaller, more manageable segments. These small segments of predetermined sizes slide across the image, with some overlap, ensuring every portion of the image is systematically evaluated. The decision to have these overlaps is intentional; it guarantees that cracks lying at the borders of two windows are not inadvertently split or missed.

In this study, a window size of 227 × 227 pixels was selected in order to match the input size of the customized MobileNetV2 model. The stride of the sliding window, which determines the step size the window takes before capturing the next segment, was set to ensure a 50% overlap between consecutive windows. This overlap was a strategic decision to enhance the likelihood of capturing cracks that might lie on the boundaries of two adjacent windows. Through this overlap, the continuity and consistency in crack detection

across the image can be ensured. Moreover, the Non-Maximum Suppression technique was implemented to accurately detect the cracks and eliminate redundancy, as shown in Figure 13.

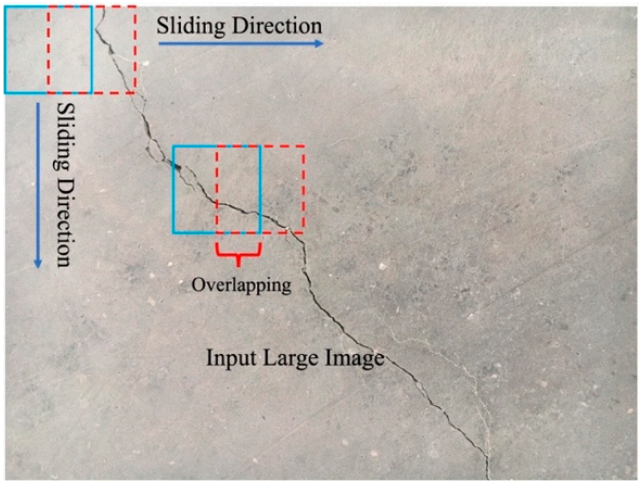


Figure 12. Sliding Windows Technique for High-Resolution Image.

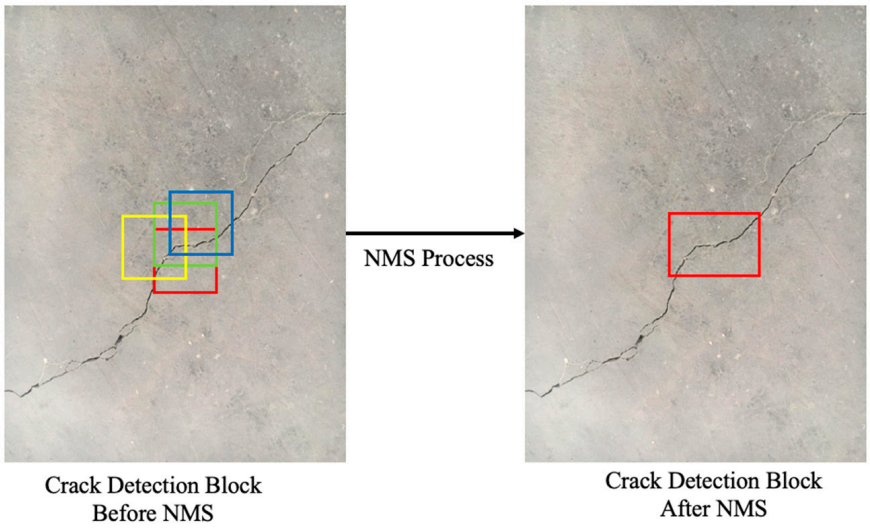


Figure 13. NMS Process to Eliminate Redundancy.

Following the application of the NMS, the focus shifted to accurately mapping the coordinates of each distinctly identified crack. An algorithmic methodology was developed to capture the specific spatial data, defined by the top-left and bottom-right vertices of the bounding boxes. This process yielded a detailed perspective on the crack's position and proved essential in the formulation of analytical reports, reinforcing the precision and consistency of the spatial data.

4.3. Width Estimation

Crack width estimation is essential to understanding the severity and potential implications of observed defects in concrete structures. In this study, an approach based on image processing and mathematical modeling was adopted to determine the width of detected cracks with a high degree of precision. This method evaluates the width of the crack based on the spatial distribution of pixel intensity values across the crack's span. The distance to the closest zero (or background) for each pixel of the crack is calculated, thereby

giving a representation of the crack's breadth at various points. The mathematic equation is given by the following:

$$D(x,y) = min_{(i,j)\in\Omega}\sqrt{(x-i)^2+(y-i)^2} \quad (4)$$

where $D(x,y)$ is the distance of a point (x,y) from the closest zero pixel in the set Ω.

From the derived distance map, the maximum distance represents half of the crack width since the distance is calculated from the center of the crack to one edge (as shown in Figure 14). Therefore, the estimated crack width in pixels is given by the following:

$$W_{in\ pixel} = \max\left(D(x,y)\right) \quad (5)$$

While the width estimation in pixel units offers a relative measurement, converting this to real-world dimensions is significant for practical applications. Given the known distance from the camera to the concrete surface, the ratio of real dimensions to pixel dimensions was calculated. The formula adopted in this study for this conversion is based on the relationship between the real-world size (width) of an object and its size on the image sensor of a camera, which is governed by similar triangles as shown in Figure 15.

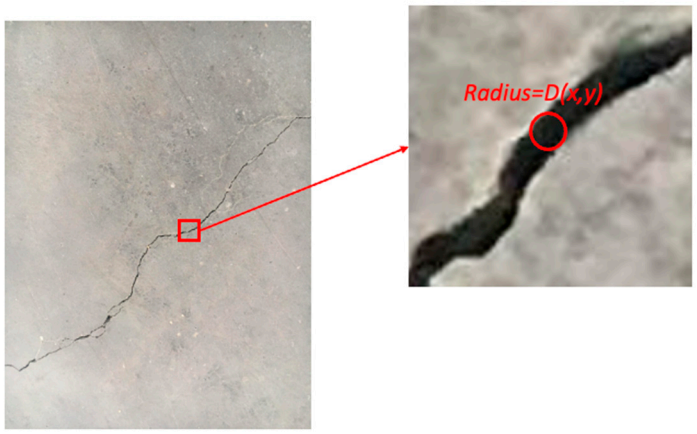


Figure 14. Crack Radius during the Calculation of Crack Width.

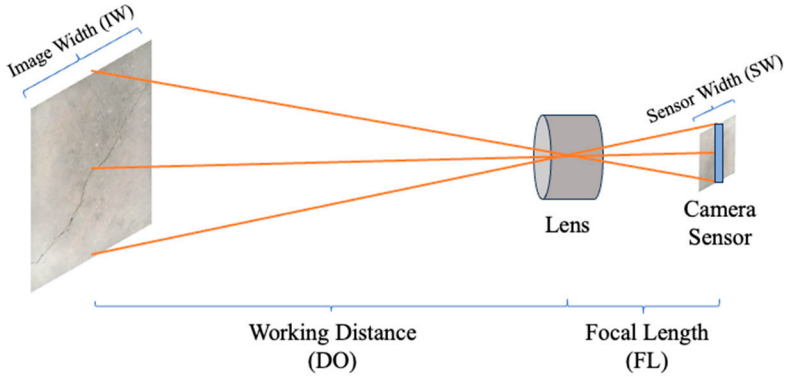


Figure 15. Relationship between Image Width, Camera Sensor, and Working Distance.

If we know the focal length of the camera, the distance from the camera to the object, and the size of the object on the image sensor, the real-world size of the object can be estimated using the following equation:

$$W_{real} = \frac{W_{pixel} \times D}{F_{pixel}} \quad (6)$$

where W_{real} is the real-world width of the crack, and W_{image} is the width of the crack as it appears in the image with the unit in pixels. D is the distance from the camera to

the concrete surface, and F_{pixel} is the focal length in pixels. The focal length in pixels is proportional to the focal length in millimeters based on the ratio of the image width in pixels to the sensor width in mm, as follows:

$$F_{pixel} = F_{mm} \times \frac{W_{sensor\ in\ pixel}}{W_{image\ in\ pixel}} \tag{7}$$

where F_{mm} is the focal length in millimeters, W_{sensor} is the sensor width and W_{image} is the image width.

It is important to note that this estimation assumes a perpendicular capture angle between the camera and the surface. If the camera is tilted, the estimation can be off, and more advanced methods involving camera calibration and pose estimation would be needed.

5. Experimental Validation

In order to evaluate the performance and accuracy of the crack detection algorithm developed in this study, experimental validation was conducted. The image used for testing was captured using a Nikon D5100 camera with a resolution of 4928 × 3264 pixels. The precision in capturing this image was ensured by gauging the distance from the camera to the concrete surface with a laser distance measure, which indicated a distance of 518 mm. The camera settings were appropriately adjusted with a focal length of 18 mm and given the image sensor's width of 23.6 mm. Figure 16 shows the original image used in the experimental validation.

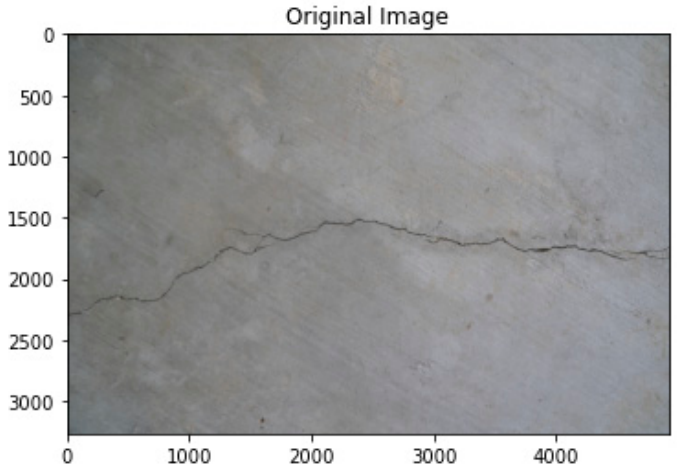


Figure 16. Original Image Used for Experimental Validation.

Considering the intricacies involved in capturing images, especially challenges due to varying lighting conditions and the textural characteristics of concrete, denoising was applied to filter out the undesired noise and improve the lighting conditions before applying the crack detection algorithm. Figure 17 shows the denoised image. This process refined the image details, enhancing the prominent features of the concrete surface, and ensuring that the actual cracks became more distinguishable.

During the evaluation of the high-resolution image, a sliding window technique was employed to systematically scan through the entire image. At each step, a window of predefined size (227 × 227 pixels) traversed the image, analyzing the portion of the image within its frame. When the model detected the presence of a crack within this window, the coordinates were noted. NMS technique was implemented to eliminate redundancy during this process. After processing the entirety of the image, these individual detections were merged to produce a comprehensive mask. This mask effectively illustrated the

overall crack pattern across the image. By highlighting these crack regions, the mask provided a visual representation of the extent and trajectory of the cracks, making it easier to discern and analyze their patterns. Figure 18 shows the sliding windows and crack location tracking process in this study.

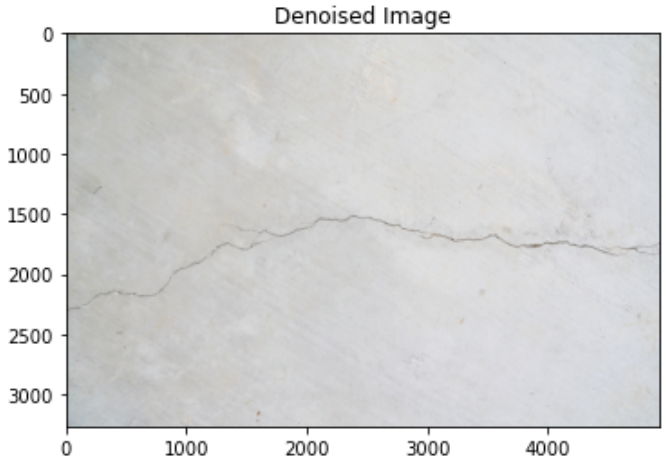


Figure 17. Image after Denoise and Lighting Adjustment.

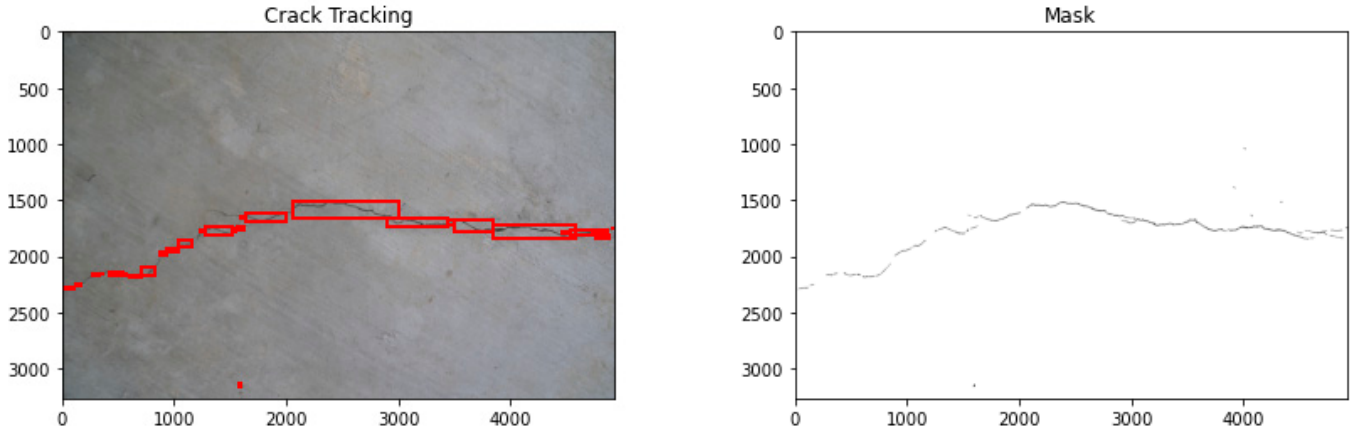
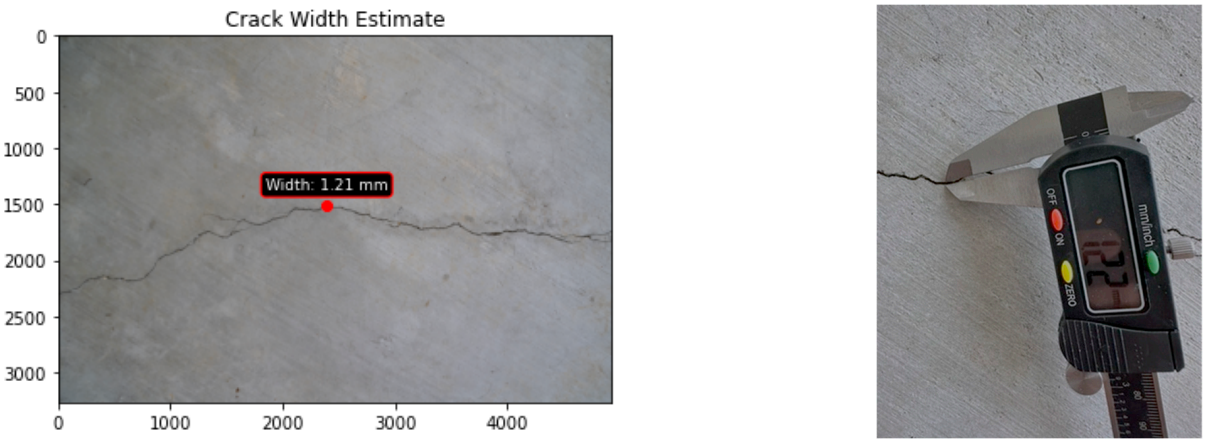


Figure 18. Crack Tracking during Sliding Windows Process and Mask Image.

The results obtained from this analytical process were presented in a visual format for clarity and better understanding. A mask image distinctly showcased the detected cracks, making it easy to visually differentiate between the damaged and undamaged regions of the concrete. Another significant visual output was an image highlighting the position of the maximum crack width, with its value labeled. The results were promising with the widest crack having a width of 8.79 pixels, which equated to a real-world width of 1.21 mm when converted using our established method. The crack width estimation results using the MobileNetV2 neural network, as shown in Figure 19a. The width was also measured as 1.22 mm using a caliper (Figure 19b), which indicated the high accuracy estimation of crack width using the algorithm developed in this study.



(**a**) Crack Width Estimate using MobileNetV2 (**b**) Crack Width Measurement

Figure 19. Crack Width Measurement using MobileNetV2 and Validation.

6. Conclusions

This study presents a comprehensive approach for concrete crack detection, highlighting the strengths of deep learning using the hybrid MobileNetV2 neural network with traditional image processing techniques. The methodology of this study leaned heavily on the MobileNetV2 neural network, which has a compact model size suitable for deployment on mobile devices. Therefore, the algorithm developed in this study can be effectively and efficiently implemented in real-world scenarios for structural concrete detection and width evaluation. Based on the findings of this study, the following conclusions can be drawn:

1. Considering the complexities of concrete textures and potential inconsistencies in image capture settings, preprocessing techniques, such as grayscale, thresholding, and normalization play an important role in enhancing the accuracy and efficiency of the neural network model.
2. The compactness of the MobileNetV2 architecture, combined with its impressive performance, reinforced its suitability for concrete crack detection and width evaluation. Compared with traditional CNN architectures, the MobileNetV2 model can potentially be effectively used in mobile devices without compromising on accuracy.
3. The sliding windows technique can efficiently localize cracks, ensuring no details are overlooked in the segmentation process.
4. The integration of camera settings with real-world metrics facilitates the conversion of pixel-level crack width into real-world dimensions. This digital-to-physical transition proves essential for practical applications and field-level implementations.
5. The experimental validation confirms that the methodology and algorithm developed in this study are effective and reliable during the concrete crack evaluation. The validation provides confidence in the application of the developed approach for practical scenarios.

While this study has made significant effects in advancing the field of concrete crack detection, there are inherent limitations that need to be addressed. The model's performance may vary under different environmental conditions, especially in low light or with varying concrete textures. Additionally, while MobileNetV2 provides the advantage of reduced model size, there could be trade-offs in terms of accuracy when compared with larger, more complex models. In addition, the method used in this study requires the distance between the camera and the concrete surface to be perpendicular when evaluating crack width. Any angular deviation might compromise the accuracy of the width measurements, as the current methodology does not account for perspective distortions induced by

angled observations. For future work, it would be beneficial to further refine the algorithm to better handle varying environmental conditions and explore the integration of other sensors for improved accuracy. Incorporating real-time feedback mechanisms for on-site adjustments and expanding the dataset to include a wider variety of concrete types and crack characteristics can also enhance the model's performance and applicability.

Author Contributions: Experimental design and methodology, L.H., A.I. and R.H.; experimental testing and data analysis, L.H. and A.I.; programming, L.H.; writing—original draft preparation, L.H.; writing—review and editing, L.H., A.I. and R.H.; project administration, R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are not available.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Safiuddin, M.; Kaish, A.A.; Woon, C.O.; Raman, S.N. Early-Age Cracking in Concrete: Causes, Consequences, Remedial Measures, and Recommendations. *Appl. Sci.* **2018**, *8*, 1730. [CrossRef]
2. Qiu, Q.; Lau, D. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. *Autom. Constr.* **2023**, *147*, 104745. [CrossRef]
3. Iraniparast, M.; Ranjbar, S.; Rahai, M.; Moghadas Nejad, F. Surface concrete cracks detection and segmentation using transfer learning and multi-resolution image processing. *Structures* **2023**, *54*, 386–398. [CrossRef]
4. Islam, M.M.M.; Kim, J.-M. Vision-Based Autonomous Crack Detection of Concrete Structures Using a Fully Convolutional Encoder–Decoder Network. *Sensors* **2019**, *19*, 4251. [CrossRef] [PubMed]
5. Yang, G.; Liu, K.; Zhang, J.; Zhao, B.; Zhao, Z.; Chen, X.; Chen, B.M. Datasets and processing methods for boosting visual inspection of civil infrastructure: A comprehensive review and algorithm comparison for crack classification, segmentation, and detection. *Constr. Build. Mater.* **2022**, *356*, 129226. [CrossRef]
6. Yalew, T.T.; Kim, K.-S. Automatic quantification of concrete cracks via multistage image filtration and trajectory-based local binarization. *J. Build. Eng.* **2023**, *77*, 107391. [CrossRef]
7. Ahila Priyadharshini, R.; Arivazhagan, S.; Arun, M. Crack recognition on concrete structures based on machine crafted and hand crafted features. *Expert. Syst. Appl.* **2023**, *228*, 120447. [CrossRef]
8. Ding, W.; Yang, H.; Yu, K.; Shu, J. Crack detection and quantification for concrete structures using UAV and transformer. *Autom. Constr.* **2023**, *152*, 104929. [CrossRef]
9. Al Biajawi, M.I.; Embong, R.; Muthusamy, K.; Ismail, N.; Johari, I. Assessing the performance of concrete made with recycled latex gloves and silicone catheter using ultrasonic pulse velocity. *Mater. Today Proc.* **2023**, *109*, S2214785323037173. [CrossRef]
10. Rahmati, M.; Toufigh, V.; Keyvan, K. Monitoring of crack healing in geopolymer concrete using a nonlinear ultrasound approach in phase-space domain. *Ultrasonics* **2023**, *134*, 107095. [CrossRef] [PubMed]
11. Candelaria, M.a.D.E.; Kee, S.-H. Evaluation of thermal damages of concrete subjected to high temperatures using recurrent neural networks for ultrasonic pulse waves. *Constr. Build. Mater.* **2023**, *407*, 133416. [CrossRef]
12. Cheng, C.C. The impact-echo response of concrete containing steel reinforcing bars, cracks around bars, and delaminations. *NDT E Int.* **1997**, *30*, 259. [CrossRef]
13. Chang, C.-C.; Yu, C.-P.; Lin, Y. Distinction between crack echoes and rebar echoes based on Morlet Wavelet Transform of impact echo signals. *NDT E Int.* **2019**, *108*, 102169. [CrossRef]
14. Rasol, M.A.; Pérez-Gracia, V.; Solla, M.; Pais, J.C.; Fernandes, F.M.; Santos, C. An experimental and numerical approach to combine Ground Penetrating Radar and computational modeling for the identification of early cracking in cement concrete pavements. *NDT E Int.* **2020**, *115*, 102293. [CrossRef]
15. Alsharqawi, M.; Dawood, T.; Abdelkhalek, S.; Abouhamad, M.; Zayed, T. Condition assessment of concrete-made structures using ground penetrating radar. *Autom. Constr.* **2022**, *144*, 104627. [CrossRef]
16. Fan, J.; Ma, T.; Zhu, Y.; Zhang, Y. Ground penetrating radar detection of buried depth of pavement internal crack in asphalt surface: A study based on multiphase heterogeneous model. *Measurement* **2023**, *221*, 113531. [CrossRef]

17. Koch, C.; Georgieva, K.; Kasireddy, V.; Akinci, B.; Fieguth, P. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Adv. Eng. Inform.* **2015**, *29*, 196–210. [CrossRef]
18. Chen, L.; Chen, W.; Wang, L.; Zhai, C.; Hu, X.; Sun, L.; Tian, Y.; Huang, X.; Jiang, L. Convolutional neural networks (CNNs)-based multi-category damage detection and recognition of high-speed rail (HSR) reinforced concrete (RC) bridges using test images. *Eng. Struct.* **2023**, *276*, 115306. [CrossRef]
19. Cha, Y.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [CrossRef]
20. Zhong, X.; Peng, X.; Yan, S.; Shen, M.; Zhai, Y. Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles. *Autom. Constr.* **2018**, *89*, 49–57. [CrossRef]
21. Nguyen, N.H.T.; Perry, S.; Bone, D.; Le, H.T.; Nguyen, T.T. Two-stage convolutional neural network for road crack detection and segmentation. *Expert. Syst. Appl.* **2021**, *186*, 115718. [CrossRef]
22. Du, T.; Chen, J.; Qu, F.; Li, C.; Zhao, H.; Xie, B.; Yuan, M.; Li, W. Degradation prediction of recycled aggregate concrete under sulphate wetting–drying cycles using BP neural network. *Structures* **2022**, *46*, 1837–1850. [CrossRef]
23. Zhang, Y.; Huang, J.; Cai, F. On Bridge Surface Crack Detection Based on an Improved YOLO v3 Algorithm. *IFAC-Pap.* **2020**, *53*, 8205–8210. [CrossRef]
24. Cui, X.; Wang, Q.; Dai, J.; Zhang, R.; Li, S. Intelligent recognition of erosion damage to concrete based on improved YOLO-v3. *Mater. Lett.* **2021**, *302*, 130363. [CrossRef]
25. Su, P.; Han, H.; Liu, M.; Yang, T.; Liu, S. MOD-YOLO: Rethinking the YOLO architecture at the level of feature information and applying it to crack detection. *Expert. Syst. Appl.* **2024**, *237*, 121346. [CrossRef]
26. Chaiyasarn, K.; Buatik, A.; Mohamad, H.; Zhou, M.; Kongsilp, S.; Poovarodom, N. Integrated pixel-level CNN-FCN crack detection via photogrammetric 3D texture mapping of concrete structures. *Autom. Constr.* **2022**, *140*, 104388. [CrossRef]
27. Li, R.; Yu, J.; Li, F.; Yang, R.; Wang, Y.; Peng, Z. Automatic bridge crack detection using Unmanned aerial vehicle and Faster R-CNN. *Constr. Build. Mater.* **2023**, *362*, 129659. [CrossRef]
28. Falaschetti, L.; Beccerica, M.; Biagetti, G.; Crippa, P.; Alessandrini, M.; Turchetti, C. A Lightweight CNN-Based Vision System for Concrete Crack Detection on a Low-Power Embedded Microcontroller Platform. *Procedia Comput. Sci.* **2022**, *207*, 3948–3956. [CrossRef]
29. Kolappan Geetha, G.; Sim, S.-H. Fast identification of concrete cracks using 1D deep learning and explainable artificial intelligence-based analysis. *Autom. Constr.* **2022**, *143*, 104572. [CrossRef]
30. Zhong, J.; Huyan, J.; Zhang, W.; Cheng, H.; Zhang, J.; Tong, Z.; Jiang, X.; Huang, B. A deeper generative adversarial network for grooved cement concrete pavement crack detection. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105808. [CrossRef]
31. Yu, Y.; Samali, B.; Rashidi, M.; Mohammadi, M.; Nguyen, T.N.; Zhang, G. Vision-based concrete crack detection using a hybrid framework considering noise effect. *J. Build. Eng.* **2022**, *61*, 105246. [CrossRef]
32. Chow, J.K.; Su, Z.; Wu, J.; Li, Z.; Tan, P.S.; Liu, K.F.; Mao, X.; Wang, Y.H. Artificial intelligence-empowered pipeline for image-based inspection of concrete structures. *Autom. Constr.* **2020**, *120*, 103372. [CrossRef]
33. Chow, J.K.; Liu, K.F.; Tan, P.S.; Su, Z.; Wu, J.; Li, Z.; Wang, Y.H. Automated defect inspection of concrete structures. *Autom. Constr.* **2021**, *132*, 103959. [CrossRef]
34. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEECVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
35. Michele, A.; Colin, V.; Santika, D.D. MobileNet Convolutional Neural Networks and Support Vector Machines for Palmprint Recognition. *Procedia Comput. Sci.* **2019**, *157*, 110–117. [CrossRef]
36. Özgenel, Ç.F. Concrete Crack Images for Classification, 2019 [Data Set]. Available online: https://data.mendeley.com/datasets/5y9wdsg2zt/2 (accessed on 10 February 2025).
37. Khudhair, Z.N.; Khdiar, A.N.; El Abbadi, N.K.; Mohamed, F.; Saba, T.; Alamri, F.S.; Rehman, A. Color to grayscale image conversion based on singular value decomposition. *IEEE Access.* **2023**, *11*, 54629–54638. [CrossRef]
38. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3rd ed.; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2007.
39. Amiriebrahimabadi, M.; Rouhi, Z.; Mansouri, N. A Comprehensive Survey of Multi-Level Thresholding Segmentation Methods for Image Processing. *Arch. Comput. Methods Eng.* **2024**, *31*, 3647–3697. [CrossRef]