

CrackGNet: A Hybrid Approach Leveraging U-Net and Graph Neural Networks for Road Crack Detection

Paras Paras

MASTER'S THESIS

A research components submitted to Auckland University of Technology in fulfilment of the
requirements for the degree of
Masters of Philosophy (M.Phil)

Supervisor

Dr. Sira Yongchareon

School of Engineering, Computer and Mathematical Sciences

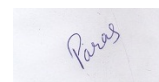
Auckland, August 2024

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor used artificial intelligence tools or generative artificial intelligence tools (unless it is clearly stated, and referenced, along with the purpose of use), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

12-08-2024

Date



Signature

Acknowledgements

I am deeply grateful to my supervisor, Dr. Sira Yongchareon, for their invaluable guidance, support, and encouragement throughout my research. Their expertise and insightful feedback were instrumental in shaping this thesis.

I am grateful to the Department Engineering, Computer, and Mathematical Sciences at Auckland University of Technology for providing me with the resources and opportunities to conduct this research. I appreciate the technical assistance received in setting up the system for this project, with special thanks to Bumjum Kim. I would also like to thank Jan Singhapan, my postgraduate coordinator, for their administrative support and guidance throughout the program.

Finally, I want to say a big thank you to my family and friends for always supporting me. A special thank you to my uncle and aunt who always believed in me. Their faith in me gave me the strength and motivation to keep going.

Auckland, August 2024

Paras Paras

Abstract

Pavement cracks are a common issue affecting road infrastructure worldwide. Their timely detection and repair are essential for ensuring road safety and preventing further deterioration. Traditional manual crack inspection methods are time-consuming, labor-intensive, and susceptible to human mistakes. To overcome these challenges, this thesis introduces CrackGNet, a hybrid deep-learning architecture for the automated detection of road cracks. Our proposed method strategically integrates a U-Net encoder-decoder with a GCN. The U-Net component focuses on robust feature extraction from crack images, while the GCN component explicitly models the spatial relationships and connectivity patterns inherent in crack structures. We performed a comprehensive evaluation of CrackGNet on the CFD, DeepCrack, and CRACK500 datasets, demonstrating its superiority over established deep learning baselines across a range of performance metrics achieving 0.8618 precision, 0.8811 recall, and an F1 score of 0.8665, outperforming existing methods. These results highlight the effectiveness of our hybrid approach and the importance of considering both local image features and spatial context in crack detection tasks. Our research contributes to the advancement of automated road infrastructure assessment systems, with the potential to improve infrastructure maintenance efficiency and reduce costs. Future research directions include exploring alternative GNN architectures for further performance gains, expanding datasets for increased model robustness, and evaluating CrackGNet in real-world inspection scenarios.

Keywords: Road Crack Segmentation, Deep Learning, U-Net, Graph Neural Networks

Contents

Attestation of Authorship	iii
Acknowledgements	v
Abstract	vii
List of Figures	2
List of Tables	3
List of Abbreviations	5
1 Introduction	7
1.1 Motivation	7
1.2 Research Justification	10
1.3 Significance and Contributions of this research	12
1.4 Thesis Structure	13
2 Related Work	15
2.1 Introduction to Road Crack Detection	15
2.1.1 The Nature and Classification of Road Cracks	16
2.1.2 Factors Underlying Road Crack Formation	18

2.1.3	The Critical Importance of Road Crack Detection	21
2.2	Evolution of Road Crack Detection Methods	22
2.2.1	Manual Inspection Methods	22
2.2.2	Traditional Image Processing Methods	23
2.2.3	Machine Learning-Based Methods	29
2.2.4	Deep Learning-Based Methods	32
2.2.5	Hybrid Approaches	39
2.3	Summary	40
3	Graph Neural Networks	41
3.1	Introduction to Graph Neural Networks	41
3.1.1	Types of Graph Neural Networks	43
3.1.2	Graph Neural Network Based Crack Detection	47
3.2	Summary	48
4	Methodology	49
4.1	CrackGNet Architecture	49
4.2	Feature Extraction using U-Net Encoder	52
4.3	Graph Construction	53
4.4	Feature Refinement using Graph Convolutional Network (GCN)	55
4.5	Feature Map Reconstruction and U-Net Decoder Fusion	57
4.6	Final Convolution	59
4.7	Loss Function	60
4.8	Summary	61
5	Experimental Results	63
5.1	Experimental Setup	63
5.2	Datasets	65

5.3	Baseline Models	67
5.4	Evaluation Metrics	69
5.5	Experimental Results	71
5.5.1	Evaluation of Results on CFD	73
5.5.2	Evaluation of Results on DeepCrack	74
5.5.3	Evaluation of Results on CRACK500	74
5.5.4	Intersection over Union (IoU) Results	75
5.6	Discussion of Overall Results	76
5.7	Discussion of Model Components	76
5.8	Summary	78
6	Conclusion and Future Work	83
6.1	Conclusion	83
6.2	Future Work	84
	Bibliography	87

List of Figures

1.1	Examples of severe pavement deterioration resulting from neglected cracks [1].	8
2.1	Longitudinal Crack [2]	16
2.2	Transverse Crack [2]	17
2.3	Alligator Cracks [2]	17
2.4	Block Crack [2]	18
2.5	Edge Crack [2]	18
2.6	Example of manual crack inspection on a bridge [3].	23
2.7	Categories of Deep Learning Methodologies for Crack Detection [4]	32
2.8	U-Net Architecture [5]	35
2.9	Segnet Architecture [6]	36
2.10	Attention U-Net Architecture [7]	37
2.11	DeepCrack Architecture [8]	38
2.12	PSPNet Architecture [9]	38
2.13	FPHBN Architecture [10]	39
4.1	Illustration of the proposed CrackGNet architecture with a detailed view of the Graph Neural Network branch.	50
4.2	A small input image with the corresponding graph overlaid (showing node and edge formation based on the distance metric)	55

5.1	Visual assessment of results from various methods on selected samples from the CFD, DeepCrack, and Crack500 datasets. The methods compared include U-Net, SegNet, Attention U-Net, DeepCrack, FPHBN, PSPNet, and the method proposed in this work.	72
5.2	Loss and IoU Curves for all datasets	73
5.3	Examples of crack detection from CFD	79
5.4	Examples of crack detection from DeepCrack	80
5.5	Examples of crack detection from CRACK500	81

List of Tables

5.1	Model hyperparameters and their values	64
5.2	CFD Dataset Results	74
5.3	Deepcrack Dataset Results	74
5.4	CRACK500 Dataset Results	75
5.5	IoU values for the three datasets.	75

List of Abbreviations

CFD Crack Forest Dataset

CNN Convolutional Neural Network

CTA Circle Transform Algorithm

GAT Graph Attention Network

GCN Graph Convolutional Network

GNN Graph Neural Network

ICA Independent Component Analysis

KNN K-Nearest Neighbours

PHT Progressive Hough Transform

SVM Support Vector Machine

CHAPTER 1

Introduction

Roads are the lifelines of our society, enabling the seamless movement of people and goods [11]. However, maintaining their structural integrity is a constant battle against harsh weather, heavy traffic, and the passage of time [12]. Cracks, which are a pervasive form of pavement damage, signal potential weakening and can escalate into costly repairs, traffic disruptions, and safety hazards [13, 14]. Traditional manual crack detection methods are slow and inconsistent while existing automated approaches often struggle with the complexities of real-world conditions. Recent advancements in deep learning offer promise for more accurate and robust crack detection. This research aims to address the limitations of existing methods, thereby contributing to the creation of improved road crack detection systems that are more effective and dependable, ultimately improving infrastructure safety and sustainability.

1.1 Motivation

The deterioration of road infrastructure poses a significant threat to safety, economic efficiency, and environmental sustainability. Proactive pavement maintenance, with accurate crack detection at its core, is essential to mitigate these consequences [15, 16]. Seemingly

minor cracks, If left unaddressed, can rapidly escalate into potholes, alligator cracking, and other severe damage (see Figure 1.1). This accelerated deterioration leads to drastically higher repair costs, traffic disruptions, and increased safety risks for all road users [13, 14].



Figure 1.1: Examples of severe pavement deterioration resulting from neglected cracks [1].

Economic Impact

Repairing road cracks can become very expensive if they are not addressed in time and develop into larger problems. A study discovered that road conditions were a significant factor in costly traffic accidents in the U.S [17]. This shows how important it is to detect road cracks early and fix them quickly. The cost of these crashes is not just a problem for government agencies, but also for taxpayers and businesses who lose money because of traffic delays and less productivity. Studies [18] and [16] have shown that doing maintenance work on time can save up to 80% of the cost of fixing roads [15]. Detecting cracks early and fixing them quickly can save a lot of money and make roads last longer.

Environmental Impact

Road cracks, which often appear first when pavement starts to fail, pose a potential risk to road safety and need to be fixed as soon as possible. If not repaired promptly, these defects

can worsen due to repeated exposure to environmental conditions or human activity [19]. These cracks also contribute to adverse environmental impacts. Studies indicate a strong link between the condition of roads and carbon emissions, with poorly maintained roads leading to higher emissions [20]. Furthermore, research conducted in [21] shows that bad roads with cracks can cause cars to use more fuel and release more harmful gases, specifically a 5% increase in fuel consumption and a 2% increase in carbon emissions.

Safety and Disruption

Unseen cracks can be very dangerous, especially for people on bikes, motorcycles, and those walking. Bad roads can cause car accidents and make driving unsafe. For example, in the UK, government data shows that 12.6% of car crashes in 2020 were caused by poor road conditions [22]. As self-driving cars become more common, it is very important to fix road surface problems to make sure these cars can operate safely [23, 24]. Besides the direct dangers, road closures or traffic problems caused by ignored cracks can lead to traffic jams, delays, and money loss for businesses and people traveling [15, 25]. Quick crack detection and repair help keep a safe and smooth road network, which is very important for a successful society.

Challenges

Despite the clear benefits of proactive crack detection, existing methods encounter technical limitations that hinder their widespread adoption in real-world scenarios. Traditional manual methods are time-consuming, labor-intensive, and prone to subjective interpretation, leading to inconsistent results. Even deep learning-based solutions often face the following challenges:

- **Fine-Grained Detail Detection:** Accurately detecting small, subtle cracks that signal early deterioration is challenging due to their limited size and low contrast within images.
- **Distinguishing Cracks from Noise:** Separating true cracks from visually similar

features such as pavement markings, stains, shadows, and other surface irregularities requires robust feature discrimination.

- **Adaptability to Real-World Variability:** Performance can degrade significantly when models are confronted with diverse lighting conditions, pavement textures, crack patterns, and environments not well-represented in training data.
- **Modeling Complex Crack Topologies:** Capturing the intricate branching patterns and long-range dependencies within crack structures is difficult using pixel-level analysis.
- **Computational Efficiency:** Depending on the model architecture, achieving high accuracy may come at the cost of computational complexity, hindering real-time or resource-constrained applications.

These shortcomings often stem from a focus on individual pixels rather than broader crack structures, topologies, and relationships.

1.2 Research Justification

Existing methods for road crack detection, which often rely on traditional image processing approaches or solely on Convolutional Neural Network (CNN)s, face several significant limitations. For instance, these methods may struggle to reliably detect fine cracks, which can be crucial to identify for timely maintenance and to prevent expensive long-term damage. The inability to detect these fine cracks can lead to their evolution into larger, more detrimental cracks, escalating the cost and complexity of road maintenance.

The conventional method of pavement assessment involves visual inspections conducted by human experts, which is considered the simplest approach [26]. While this method can yield reasonably accurate results, it has several drawbacks. These include disruption to

traffic, subjective evaluations, risks to on-site inspectors, significant costs, and suboptimal performance [15, 18, 25].

Moreover traditional methods [27–49] can be highly sensitive to noise, complex backgrounds, varying lighting conditions, and the presence of non-crack features that visually resemble cracks. This sensitivity often leads to a lack of robustness, resulting in false positives and missed detections, which can hinder reliable crack assessment.

Although, while CNN-based methods [5–10] have brought improvements to crack detection, they primarily focus on local image features. This focus can make it challenging to model complex crack topologies and long-range dependencies, particularly in challenging real-world scenarios. For instance, a crack that spans across a large section of the road might be detected as multiple smaller, unconnected cracks, leading to an inaccurate representation of the road’s condition.

This research aims to directly address these limitations by introducing a hybrid architecture that combines U-Net, a type of CNN, and Graph Neural Network (GNN). This unique integration leverages the strengths of GNNs [50] in modeling structural relationships, with the aim of enhancing detection accuracy and robustness. For example, a GNN can model the relationship between different parts of a large crack, leading to a more accurate detection and representation of the crack.

The hybrid architecture also has the potential to improve computational efficiency. By using GNNs to focus on the relationships between key areas within the image, we reduce the need for exhaustive processing of the entire image. This targeted analysis is a critical aspect, making the architecture a promising avenue for real-world crack detection applications where real-time or resource-constrained solutions are necessary.

By addressing the limitations of existing methods and introducing a hybrid approach, this research contributes to the ongoing efforts to enhance road safety and streamline maintenance processes. The proposed hybrid approach enhances the accuracy and reliability of crack de-

tection and also provides a foundation for further research into the theoretical understanding of crack characteristics.

1.3 Significance and Contributions of this research

This research aims to make the following significant contributions to the domain of road crack detection:

- **Innovative Methodology:** The development of a hybrid deep learning architecture that combines the capabilities of U-Net's feature extraction with the structural modeling power of GNNs. This unique integration is designed to enhance the accuracy and dependability of road crack detection, addressing the limitations of existing methods.
- **Enhanced Fine-Crack Detection:** The proposed approach demonstrates an improved ability to detect fine cracks, which are often challenging for traditional methods. This enhanced detection is crucial for timely maintenance intervention and prevention of further deterioration, contributing to improved road safety and maintenance efficiency.
- **Increased Robustness and Reliability:** The hybrid model exhibits greater adaptability to real-world challenges such as changing lighting, complex textures, and the presence of visually similar non-crack features. This adaptability results in a more robust and reliable crack detection system, reducing the occurrence of false positives and missed detections.
- **Modeling of Long-Range Dependencies:** The proposed approach leverages the power of GNNs to model long-range dependencies within crack patterns. This capability enhances the detection of complex crack structures and reduces false positives due to localized noise or visually similar non-crack features, thereby refining the overall accuracy and reliability of the crack detection system.

- **Potential for Improved Computational Efficiency:** The integration of GNNs in the proposed model possesses the potential to reduce computational overhead by focusing on crack structure and relationships. This potential improvement in computational efficiency makes the proposed model a promising solution for real-time or resource-constrained applications.
- **Advancement of Theoretical Knowledge:** The use of GNNs in the proposed model provides a deeper understanding of crack patterns in road images. This improved understanding could help us research road cracks more effectively and might lead to better detection techniques in the future.

1.4 Thesis Structure

Chapter 2 provides a thorough review of the literature on road crack detection approaches, highlighting their strengths and limitations. Chapter 3 presents the detailed methodology of the proposed system, including image acquisition techniques, pre-processing steps, the deep learning model architecture, and training procedures. Chapter 4 presents the results of the experiments conducted on the proposed model, encompassing both quantitative and qualitative assessments. Chapter 5 offers a thorough discussion of these results, interpreting their significance and exploring broader implications.

CHAPTER 2

Related Work

Road crack detection plays a dominant role in proactive pavement preservation, safeguarding the safety and efficiency of transportation networks. Accurate and reliable identification of cracks is crucial for minimizing the economic and environmental consequences of neglected deterioration. This literature review explores the evolution of crack detection methods, starting with manual inspection techniques and progressing to automated image processing approaches to the transformative power of deep learning. Special emphasis will be placed on deep learning architectures, their potential for addressing complex crack patterns and challenging real-world conditions, and how hybrid approaches offer further advancements. This analysis will highlight the gaps and limitations in existing methods, setting the stage for the proposed work.

2.1 Introduction to Road Crack Detection

Road networks constitute critical infrastructure assets that underpin economic prosperity, transportation efficiency, and social well-being. However, the longevity of these networks is continually challenged by various forms of deterioration, with cracking representing one of

the most pervasive and detrimental distresses. Timely and accurate detection of road cracks stands as an indispensable cornerstone of pavement asset management, safeguarding user safety, ensuring uninterrupted service, and promoting cost-effective maintenance practices.

2.1.1 The Nature and Classification of Road Cracks

Road cracks represent more than mere surface-level imperfections; they serve as indicators of the underlying structural health of the pavement system. To develop effective diagnostic and repair strategies, it is essential to understand the various manifestations of cracks. Common classifications [2] include:

- **Longitudinal Cracks:** These extend parallel to the direction of traffic flow and frequently signal fatigue-induced distress within the pavement structure or potential issues in underlying pavement layers. Longitudinal cracks may also arise from inadequacies in longitudinal joint construction.



Figure 2.1: Longitudinal Crack [2]

- **Transverse Cracks:** These cracks intersect the road's centerline perpendicularly. Thermal stresses, caused by temperature-induced expansion and contraction, are primary

contributors to transverse crack formation. Shrinkage of asphalt materials or reflective cracking emanating from lower concrete layers can also precipitate this crack type.



Figure 2.2: Transverse Crack [2]

- **Alligator Cracks:** This type of cracking, which looks like a network of interconnected cracks similar to alligator skin, indicates serious structural breakdown within the pavement system. Common causes encompass fatigue from repeated heavy loads, insufficient pavement thickness, or moisture-induced damage within the base or subbase layers.



(a) Moderate Alligator Crack (b) Severe Alligator Crack

Figure 2.3: Alligator Cracks [2]

- **Block Cracks:** Forming rectangular or square patterns, block cracks frequently stem from asphalt shrinkage during the curing process or inadequate accommodation of thermal expansion and contraction. While generally not indicative of severe structural deficiencies, they facilitate water infiltration.



Figure 2.4: Block Crack [2]

- **Edge Cracks:** These cracks manifest along the periphery of the pavement. Primary causes include insufficient shoulder support, weakening of the pavement edge due to moisture intrusion, or root growth from adjacent vegetation.



Figure 2.5: Edge Crack [2]

2.1.2 Factors Underlying Road Crack Formation

The emergence of road cracks stems from a multifaceted interplay of factors that progressively degrade the integrity of the pavement system. A thorough understanding of these factors is

crucial for effective road design, construction, and maintenance practices, leading to increased road longevity and reduced maintenance costs.

- **Traffic Loading:**

- **Traffic Volume and Weight:** Repetitive loading from vehicles induces fatigue and strain within the pavement structure. Heavier vehicles, higher traffic volumes, and factors like tire pressure, axle configurations, and unbalanced loads significantly exacerbate crack development [51].
- **Traffic Patterns:** Uneven traffic distribution, particularly concentrated loads towards lane edges, can accelerate crack initiation at those specific locations [52].
- **Traffic Growth:** Failure to consider future traffic projections during design can lead to premature pavement failure due to increased loading exceeding the initial design capacity [53].

- **Environmental Conditions:**

- **Climate:**

1. Repeated freezing and thawing of water within the pavement structure can cause expansion and contraction, leading to cracking. The severity of this effect depends on factors like the periodicity of freeze-thaw cycles, water content within the pavement, and the specific type of aggregate used [54].
2. Wide temperature variations can cause thermal stresses within the pavement, leading to cracking. This is particularly relevant in regions with significant seasonal temperature changes [55].
3. Moisture infiltration into the pavement structure weakens the underlying materials and reduces the load-bearing capacity. Additionally, trapped moisture can exacerbate freeze-thaw damage and contribute to the formation of potholes [56].

– **Geologic Conditions:**

1. Subgrade characteristics significantly influence pavement performance. Expansive clays can place additional stress on the pavement due to seasonal volume changes. Similarly, poor drainage in the subgrade can lead to moisture accumulation and subsequent weakening of the foundation [57].
2. Areas prone to earthquakes experience additional dynamic loads that can accelerate pavement deterioration and crack formation [58, 59].

• **Construction and Material Deficiencies:**

- **Pavement Design:** Inadequate design considerations for anticipated traffic loads, environmental stresses, and subgrade conditions can result in pavements susceptible to cracking [60].

– **Material Quality:**

1. Substandard asphalt mixtures lacking sufficient strength, flexibility, or resistance to fatigue and moisture damage can lead to premature cracking [61].
2. The size, shape, and quality of aggregates used in both the asphalt mix and the base/subbase layers influence the overall strength and durability of the pavement [62].

- **Construction Practices:** Improper construction techniques can introduce vulnerabilities and expedite crack formation. Examples include:

1. Inadequate compaction leading to weak and unstable pavement layers [63].
2. Poor drainage design or construction flaws leading to water accumulation within the pavement structure [64].
3. Incorrect placement or sealing of joints allowing water infiltration and weakening of the underlying materials [63, 65].

2.1.3 The Critical Importance of Road Crack Detection

Proactive crack detection extends beyond a mere best practice, it serves as the cornerstone of effective and sustainable pavement management strategies. Timely and accurate crack data empowers informed decision-making, yielding a multitude of long-term benefits:

- **Enhanced Maintenance Prioritization:** Budgetary constraints often necessitate strategic allocation of maintenance resources. Comprehensive crack data facilitates the objective identification of road segments in the most critical need of repair, maximizing the return on investment for maintenance activities [66].
- **Optimized Repair Techniques:** Employing generic approaches to pavement repair proves inefficient and frequently ineffective. A thorough understanding of the crack's nature, extent, and severity is crucial for selecting the most appropriate and long-lasting rehabilitation techniques, ultimately preventing premature failures and extending pavement service life [67].
- **Improved Safety Assurance:** Cracks, particularly those classified as severe, pose a significant safety hazard to all road users. Early detection empowers road authorities to implement timely measures to address these safety concerns, minimizing the potential for accidents and safeguarding public well-being [15, 25].
- **Cost-Effective Pavement Management:** Proactive crack treatment, especially in the early stages of crack development, represents a strategic investment in infrastructure longevity. By preventing the progression of minor cracks into major failures, road agencies can significantly reduce long-term maintenance and reconstruction costs, leading to substantial financial savings [16, 18].
- **Data-Driven Pavement Performance Monitoring:** Regular and systematic crack detection establishes a historical dataset that tracks pavement conditions over time.

This data becomes invaluable for assessing the effectiveness of implemented maintenance strategies and informing future advancements in pavement design and material selection [68]. Through continuous data analysis, pavement management practices can be continuously refined for optimal performance.

2.2 Evolution of Road Crack Detection Methods

2.2.1 Manual Inspection Methods

Road maintenance has traditionally relied on manual inspection methods to identify and assess cracks. In this approach, trained inspectors physically examine road surfaces, searching for visible signs of deterioration such as cracks, fissures, or other irregularities. The primary method is visual inspection, where inspectors directly scan the pavement for defects. While conceptually simple, manual inspection faces several inherent challenges. The accuracy of manual inspections is significantly impacted by the inspector's expertise, skill level, and visual perception [69]. This variability can lead to inconsistencies in crack detection and classification. Additionally, manual inspection, particularly for extensive road networks, is a time-consuming and labor-intensive task [70]. While simple tools like straightedges and probes can augment visual inspection, they offer only incremental improvements in precision.

The limitations of manual road crack detection, including subjectivity, traffic disruption, risks to inspectors, high costs, and potential for suboptimal performance, have underscored the need for more objective, efficient, and scalable assessment techniques [15, 18, 25, 71]. Traditional image processing methods offer a significant step forward by automating aspects of crack detection, addressing the shortcomings of manual approaches.



Figure 2.6: Example of manual crack inspection on a bridge [3].

2.2.2 Traditional Image Processing Methods

In the early stages of detecting road cracks through computer vision, traditional image processing methods played a dominant role. These methods focused on the direct manipulation of image pixels to identify and highlight cracks. Techniques such as pixel intensity manipulation, edge detection, and the use of hand-designed feature extractors were common.

Thresholding Based Methods

Thresholding is a time-honored and straightforward technique used to segment an image into similar regions [72]. It plays a pivotal role in crack detection systems as it enables the classification of image pixels into two groups based on their gray values: ‘crack’ and ‘non-crack’. Pixels whose intensity values surpass a predetermined threshold are labeled as cracks, whereas those with intensity values at or below the threshold are classified as non-crack pixels. This method is known as bilevel thresholding. In a more sophisticated approach, several thresholds can be chosen, partitioning the complete range of gray values into multiple subranges. This technique is referred to as multilevel thresholding [73].

The thresholding method is widely used in pavement engineering and management systems, specifically in the examination of images of cracks [74]. The result of the thresholding

procedure transforms the initial grayscale image into a binary image, in which pixels with a value of 0 belong to the ‘non-crack’ category, while pixels with a value of 1 signify ‘crack’. Acquiring a binary image early on is helpful as it reduces the complexity of the information, making the recognition and classification process more manageable [75]. Equation 2.1 provides a clear definition of a thresholded image $g(x,y)$, where:

- x and y denote the location of a pixel within the image.
- T is the threshold value that distinguishes whether a pixel corresponds to a foreground (crack) or background(non-crack).

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases} \quad (2.1)$$

The main difficulty with this approach is choosing a suitable threshold value. For optimal crack detection and noise reduction in the results, it is recommended to develop an adaptable rule set that can adjust to variations in input images [76]. Over the course of the last forty years, numerous strategies for determining the appropriate threshold value have been suggested, aiding in the development and enhancement of this basic crack detection method.

Koutsopoulos et al. [77] proposed a thresholding method based on histogram that outperforms basic histogram thresholding by considering spatial relationships between pixels. However, the iterative method can still lead to poorly segmented and hard-to-identify distresses. Koutsopoulos and Downey [78] introduced a multiple thresholding approach, demonstrating effectiveness with block cracks but struggling to accurately capture faint horizontal cracks and introducing fragmentation in alligator crack patterns.

Koutsopoulos et al. [79] presented several additional thresholding methods, including Otsu, Kittler, Regression-based, and Relaxation, with relaxation and regression-based techniques

proving superior. Regression-based offers faster computation, while relaxation provides less noisy results. Oh et al. [80] proposed a method based on iterated clipping, performing well under uniform lighting but suffering accuracy losses when shadows are present. Chen et al. [81] proposed the OtsuNiblack adaptive thresholding method to address issues with shadows, outperforming the Otsu method but potentially requiring pre-processing to filter noise and shadows.

Oliveira and Correia [82] introduced another thresholding approach, offering good crack detection and classification while sometimes still experiencing accuracy issues and often requiring pre-processing steps. Tsai et al. [83] developed a dynamic (adaptive) thresholding method utilizing optimization strategies, generally outperforming other methods and demonstrating robustness across challenging conditions with varying lighting, shadows, and noise. Salari and Bao [84] applied a relaxation method for crack segmentation, successfully removing unwanted fragments and filling missed crack sections with relative simplicity in implementation. However, this method struggles under varying lighting and surface conditions, particularly where low contrast exists amidst the background and cracks.

Salari and Ouyang [85] introduced a modified fractal thresholding approach, reporting high segmentation accuracy, but details of the method are not available from the image. Xu et al. [86] employed the Otsu algorithm combined with morphological filtering, though the table lacks specific results or limitations for this method. Lokeshwor et al. [87] utilized adaptive thresholding, noted for its speed and outperforming other tested methods, though it requires both pre- and post-processing techniques. Song et al. [88] adopted a double threshold approach, excelling in identifying small cracks and noise reduction. Liu [89] focused on local/adaptive thresholding, which proved effective in extracting cracks from images with shadows and offered accurate positioning.

Akagic et al. [30] modified the traditional Otsu method, achieving high accuracy even in low signal-to-noise environments but requiring a post-processing stage to handle the remaining

noise. Hu et al. [90] used multi-threshold segmentation, demonstrating positive performance metrics, but the image lacks information on specific limitations. Safaei et al. [91] presented a modified dynamic thresholding approach suitable for noisy environments, offering speed and potential value in resource-limited projects.

Edge Detection Based Methods

Apart from thresholding methods, edge-detection algorithms have become increasingly popular in the realm of image processing, often being used for crack segmentation. Edges in digital images are characterized by changes in light, color, and texture. These variations in intensity can be utilized to derive features from the image [92]. The goal of edge-based algorithms is to delineate objects or regions within an image by detecting variations in pixel values [93].

Edge detection methods are very different from those based on thresholds. These edge-based methods look at straight-line features that mainly match up with the edges of cracks and other important details on the surface of the object. An edge is like a border between two areas that have different shades of gray. It's thought that the shift from one of these areas to the other can be spotted just by looking at changes in these shades of gray [94].

The edges of objects often lead to sudden shifts in the brightness of an image. Also, different parts of an object soak up varying amounts of light, causing changes in brightness. So, the information about brightness changes in an image likely points to the boundaries of objects [95].

Due to the visual similarity between pavement cracks and edges, edge detection methods like Sobel and Canny [96] are used for crack detection. Canny is a widely used algorithm that involves the following steps: first, a 5×5 Gaussian filter is applied to reduce noise and smooth the image. Subsequently, Sobel kernels are employed to calculate the horizontal (G_x) and vertical (G_y) gradients of each pixel. These gradients then allow for the calculation of the edge direction and gradient using the following equations:

$$EdgeGradient(G) = \sqrt{G_x^2 + G_y^2} \quad (2.2)$$

$$Angle(\theta) = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.3)$$

After calculating gradients, the Canny Edge Detection algorithm looks at each pixel to see if its gradient size is the biggest in its area, which helps to make potential edges thinner. Then, it uses two limits (high and low) to sort edges: pixels with gradients above the high limit are definitely edges, while those below the low limit are thrown away. Pixels with gradients between the limits are seen as edges if they're connected to ones that are definitely classified, making sure there's continuity. This mix of looking at local maxima and using limits with hysteresis lets the algorithm effectively guess the real edges in an image.

Various edge detection techniques have been adapted for crack detection. Li [33], for example, combined an eight-directional Sobel operator with preprocessing steps – neighborhood weighted averaging for noise reduction and Otsu's algorithm for segmentation – to improve accuracy. Santhi et al. [34] successfully employed the Canny operator in conjunction with a Butterworth filter for pavement crack detection. Their approach begins by converting the RGB image to grayscale. To minimize the impact of noise, a Butterworth filter is applied; this filter's threshold adapts dynamically based on the image's intensity levels. The threshold is determined by the intensity contrast distinguishing the foreground (cracks) from the background (pavement). Filtering the image in this manner yields a clear representation of the cracks, suitable for the application of the Canny edge detection algorithm.

Feature Extraction Based Methods

Feature extraction techniques are pivotal in image processing, especially in tasks like crack detection. Methods analyzing image texture, like wavelet transforms, have demonstrated superior reliability and accuracy over non-texture-based methods such as Canny, Sobel, or Fourier transforms [96–99]. However, edge detection techniques that utilize Sobel and Laplacian operators offer superior computational speed for identifying patterns such as cracks [100]. Notably, for measuring crack widths Laplacian methods have been employed.

In cases of complex crack distributions, especially those occurring alongside linear road markings, Circle Transform Algorithm (CTA) and Progressive Hough Transform (PHT) have proven effective in preventing the misidentification of non-crack linear features as cracks [101].

Road cracks typically exhibit different intensity levels compared to the background. Consequently, determining the average and spread of pixel intensities can help distinguish them from surrounding pixels [48]. Integrating these features with shape, orientation, and continuity features can improve noise reduction and the recognition of continuous crack patterns [102].

Yamaguchi and Hashimoto leveraged inter-pixel connectivity features for concrete surface crack detection [35]. They employed a percolation process, utilizing the circularity of connected pixels, to assess whether these pixels formed a crack. This method could be replaced with template matching to detect more intricate patterns [36].

The use of Gabor filters has been implemented for the analysis of crack orientation details [37]. After applying a Gaussian filter at different angles, the strongest response at each pixel can help classify the image [38]. Prominent image features like contrast and rarity can be utilized to distinguish cracks, given that these specific attributes are less prevalent and less uniformly distributed compared to the overall background texture of the image [39].

However, current methods mainly focus on individual pixel features, which can hinder overall computational efficiency. Moreover, these methods' accuracy can be compromised

by interferences like stains, shadows, and rough surface conditions, often referred to as noise [39].

These methods may struggle with accuracy under changing lighting or weather conditions. To overcome these challenges, techniques like Independent Component Analysis (ICA) have been explored to handle different levels of dampness in structures [40]. Monitoring feature movement in videos has been employed to account for changing lighting conditions when identifying stress cracks on steel bridges [41]. Additionally, back-end image recognition software has been developed for the automatic identification of road damage, converting observations into a pavement condition index value indicative of road quality [42].

In conclusion, while traditional feature extraction methods have made significant strides in crack detection, they have limitations. These include a focus on local features, susceptibility to noise, and challenges under irregular illumination conditions. However, the advent of machine learning techniques offers promising avenues for overcoming these limitations. In the following section, we will explore machine learning-based techniques, exploring how they can improve the accuracy and efficiency of identifying and categorizing cracks. [42].

2.2.3 Machine Learning-Based Methods

The limitations of traditional image processing techniques in handling complex real-world crack detection scenarios have led to the exploration of machine learning algorithms. Machine Learning approaches offer the ability to learn complex patterns from data, overcoming the need for manually designed features.

Supervised Learning

Supervised learning methods leverage explicit annotations to identify whether the input image contains crack patterns. Techniques such as Support Vector Machine (SVM)s [24, 103–106] have been employed in studies like [43] to differentiate between normal and cracked road

surfaces within selected image regions. Similarly, other neural network architectures have been applied to the task of crack detection such as [44–46].

These approaches initiate the process by dividing the input image into smaller, more manageable sections known as patches. Next, features that are meticulously crafted to identify and represent the unique characteristics of cracks are extracted from each individual patch. The characteristics derived from these procedures are subsequently utilized as the foundation for training algorithms in machine learning. These algorithms are ultimately employed to locate the presence of cracks on road surfaces.

Although these approaches have shown superior performance compared to basic image-processing-based methods, they are not without their limitations. The employment of hand-crafted features may limit the effectiveness of these approaches when dealing with changes in lighting caused by weather or time of day. Additionally, these approaches may struggle to encompass complex feature distributions due to the specificity of hand-crafted features, which may not generalize well across different tasks or conditions.

Unsupervised Learning

Unsupervised learning methods aim to detect road cracks without the need for explicit annotations. These methods construct a model based on the features representing normal road surfaces, as seen in [47–49, 82, 107]. The model operates under the assumption that its training data consists solely of images depicting normal road surfaces. Therefore, any deviation observed in the model’s output from this established baseline is interpreted as an indication of a potential crack.

The success of these methods depends heavily on the quality of the underlying probability model. Techniques such as kernel density estimation have been employed to construct a probabilistic representation of the typical features observed in images of undamaged road surfaces. [47], [82]. Similarly, studies like [48] and [49] have proposed unsupervised road

crack detection methods founded on the Gaussian mixture model and k-means clustering, respectively.

While these techniques can be helpful for detecting cracks without the need for manual labeling, they may struggle to generalize when faced with the diverse and intricate conditions of actual roads. This is often due to a tendency to overfit the specific characteristics of the training data. Furthermore, the underlying assumption that features follow a Gaussian or linear distribution can be a significant limitation. When the underlying data distribution differs from this assumption, the effectiveness and accuracy of these methods may be compromised.

Consequently, ensuring the effectiveness of crack detection methods necessitates careful consideration of probability models and thorough validation of their performance on diverse real-world datasets. Exploring methods that make fewer assumptions about the data distribution, such as non-parametric methods or deep learning techniques, which can automatically learn the data distribution from the data itself, could also be beneficial. However, these methods also present their own set of challenges, such as the necessity for extensive datasets and substantial computational resources.

In conclusion, both supervised and unsupervised learning approaches hold the potential for road crack detection but face inherent limitations. These methods often struggle with the complexity of real-world pavement images due to their reliance on shallow learning. These techniques cannot adequately extract and interpret higher-level features or handle variations in illumination and environment.

To address these limitations and enhance road crack detection, we must turn to more sophisticated techniques. This leads us naturally into the realm of deep learning, where models can automatically learn complex features from data. Deep learning offers the potential for greater robustness, adaptability, and improved performance. However, these methods also introduce their own unique challenges, which we will explore in the next section.

2.2.4 Deep Learning-Based Methods

Deep learning [108], specifically CNNs, has revolutionized computer vision, and road crack detection has significantly benefited. CNNs excel at autonomously learning hierarchical image representations, eliminating the need for hand-crafted features and surpassing the performance of previous machine-learning techniques. Deep learning crack detection falls into three main categories(see Fig 2.7) :

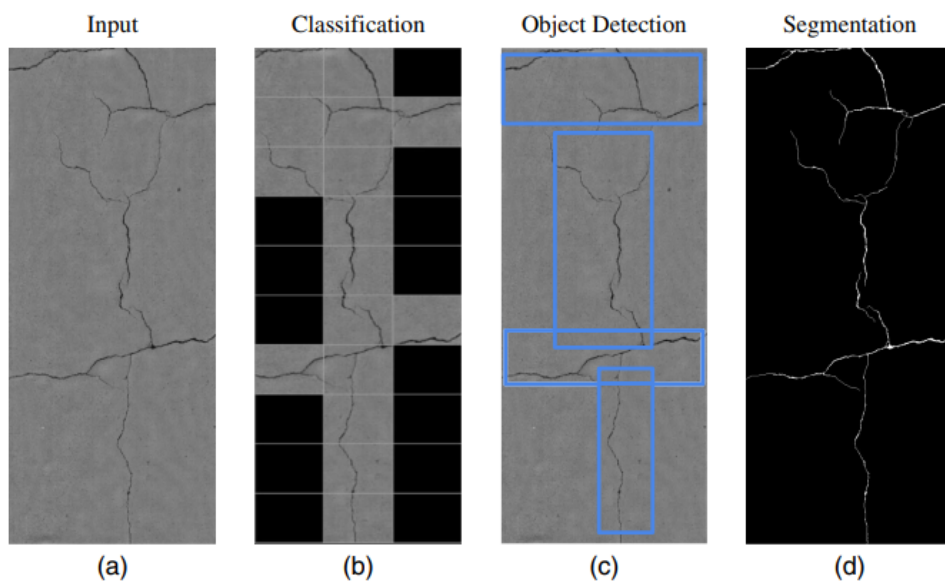


Figure 2.7: Categories of Deep Learning Methodologies for Crack Detection [4]

Classification

Classification aims to identify the occurrence of cracks in an image/image patch and then categorize them (see Fig 2.7(b)). Deep learning, first applied to crack classification in 2016 [109, 110], rapidly became the most preferred deep learning method by 2017. CNN models that incorporate fully connected layers are the standard, often utilizing lightweight models (not more than 10 layers) due to the binary nature of the core classification task.

Classification is divided into two main approaches:

- **Image-Level Classification:** A deep learning model analyzes an entire image at once to determine if cracks are present. Several researchers have explored this approach [111–114], employing deep learning models that analyze complete images from the dataset to detect cracks.
- **Patch-Level Classification:** Patch-level classification is far more popular. The process involves breaking down the original image into smaller patches and then individually analyzing each patch with the deep learning model to determine the presence of cracks. This method offers several advantages: it generates significantly more training data (improving model accuracy), provides a general idea of where cracks are within the original image, and the localization data can be used to identify crack types and patterns. Numerous studies [109, 110, 115–125] have focused on patch-level classification due to its advantages in crack detection.

While both image-level and patch-level classification offer advantages, they have limitations in terms of the precision and detail they provide about cracks. These methods may struggle to accurately outline crack length, width, and complex branching patterns, which are important for engineering and research purposes.

Object Detection

Object detection tasks [115, 126–130] identify cracks by drawing bounding boxes around them (see Fig 2.7(c)). These models can also classify the type of crack found within each box. Existing object detection models like Faster-RCNN [131] and SSD [132] are commonly used, with researchers experimenting on different backbone networks, such as VGG [133] and ResNet [134] and training strategies for optimal results.

Like patch-level classification, object detection identifies the precise coordinates and extent of cracks. However, the generated bounding boxes provide limited detail and cannot accurately represent the complex shapes of cracks. This makes it impossible to directly estimate

important crack features such as length, width, and branching patterns.

Segmentation

Segmentation provides the most detailed crack maps. It works by labeling every single pixel in an image to either ‘crack’ or ‘not crack’ (see Fig 2.7(d)). This precise information about crack location and shape is incredibly valuable. Not only does it help identify the crack type, but it lets engineers and researchers directly measure important details like length, width, and complex branching patterns. Advances in technology mean we now have high-resolution 2D and 3D image data, making detailed segmentation even more feasible. This combination of factors makes segmentation the most popular and cutting-edge approach for deep learning crack detection.

Deep Learning Methods for Crack Segmentation

While deep learning offers several approaches to crack segmentation, the most prevalent approach involves utilizing and modifying encoder-decoder-based models. Researchers have extensively explored and adapted various encoder-decoder architectures for crack segmentation, including:

U-Net [5] a popular encoder-decoder architecture (see Fig 2.8), has become a popular choice for crack segmentation [135–137]. Its power lies in its encoder-decoder structure and its signature skip connections. The encoder progressively extracts features from the input image, reducing it to a compact representation, while the decoder reconstructs a detailed segmentation map, pinpointing crack locations on a pixel level. Importantly, U-Net’s architecture features skip connections that directly link corresponding layers between the encoder and decoder. This preserves fine-grained details from earlier encoding stages that are essential for precise crack localization and are often lost in traditional encoder-decoder models. By merging these low-level details with high-level semantic information captured deeper in the network,

U-Net generates remarkably accurate and nuanced crack segmentation. However, U-Net can sometimes struggle to prioritize the most relevant features and may have difficulty handling complex object scales, potentially leading to noise sensitivity and imprecise boundaries. These limitations could be addressed by incorporating a GNN into U-Net architecture, which is the focus of our proposed approach.

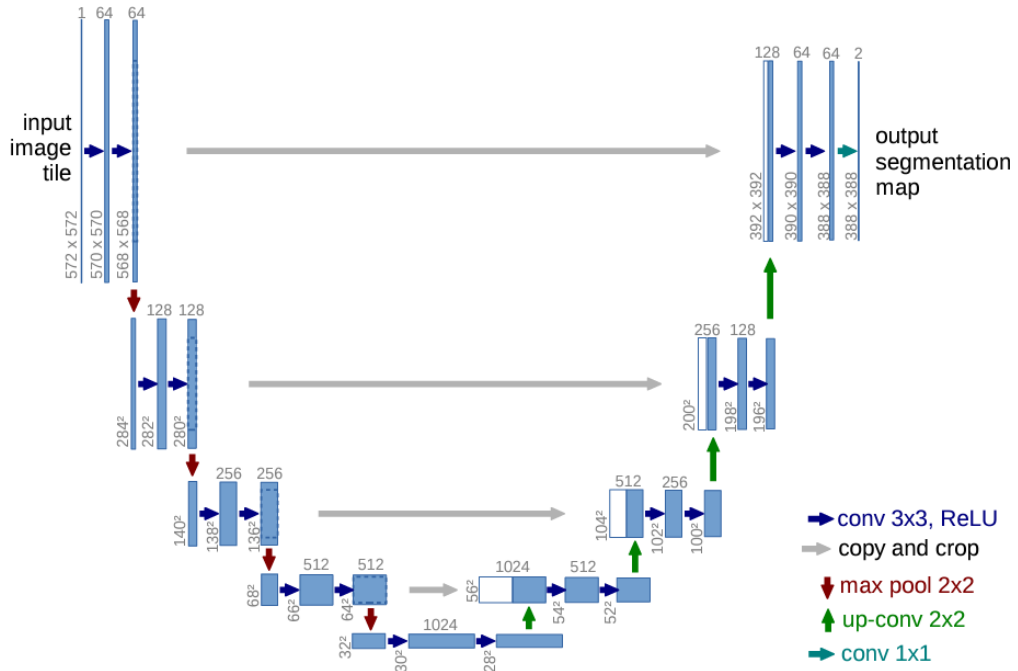


Figure 2.8: U-Net Architecture [5]

SegNet [6] is a specialized encoder-decoder architecture (see Fig 2.9) designed for efficient image segmentation tasks, including crack detection. It leverages the powerful feature extraction abilities of the VGG-16 [133] as its encoder, discarding the fully connected layers for segmentation optimization. SegNet's defining characteristic is its innovative use of max-pooling indices during the upsampling stage. These indices, recorded during the encoding process, guide the sparse upsampling in the decoder. This approach improves boundary detail reconstruction (crucial for accurate crack segmentation) compared to traditional up-

sampling methods. Additionally, SegNet’s design prioritizes computational efficiency. Utilizing index-guided upsampling, diminishes the parameter count compared to fully trainable decoder layers, making it suitable for memory-constrained environments. However, SegNet’s reliance on pooling indices for upsampling can sometimes lead to less precise segmentation, a limitation that could be mitigated by the use of a GNN for more accurate feature extraction and localization.

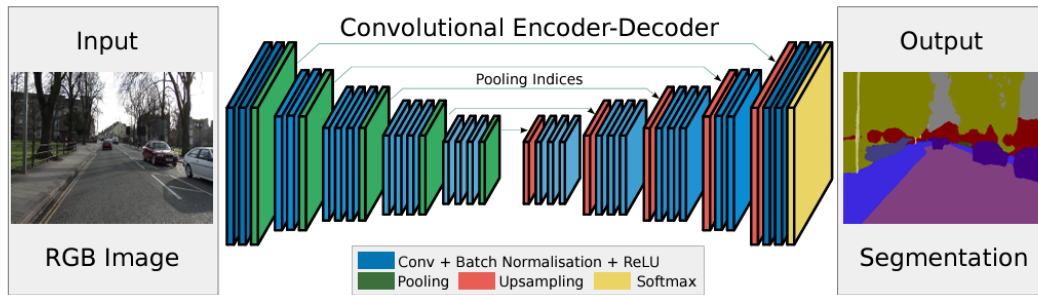


Figure 2.9: Segnet Architecture [6]

Attention U-Net [7] builds upon the foundation of the standard U-Net architecture (see Fig 2.10), strategically incorporating attention mechanisms to enhance crack segmentation performance. Its hallmark feature is the addition of attention gates within the skip connections that bridge the encoder and decoder. These gates learn to focus on the most relevant feature maps, suppressing irrelevant information and ensuring only the most crucial details are passed on for precise crack segmentation. This focus on salient details helps refine localization accuracy, allowing Attention U-Net to produce segmentation maps with sharper boundaries and a clearer distinction between true cracks and background textures. Additionally, Attention U-Net’s flexible attention mechanism enables the model to adaptively refine its focus based on the specific characteristics and complexity of the crack patterns present in an image. However, the attention mechanism may not always be able to effectively differentiate between crack features and noise, especially in complex or cluttered images. This is another area where our proposed U-Net+GNN approach could offer improvements, as GNNs are adept at capturing

intricate patterns and relationships within the data.

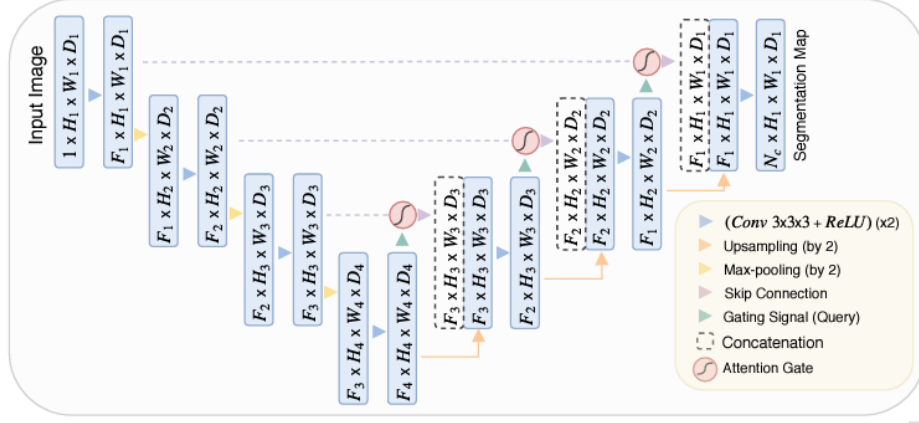


Figure 2.10: Attention U-Net Architecture [7]

DeepCrack [8] introduces a unique approach to crack segmentation by harnessing the power of multi-scale predictions (see Fig 2.11). It leverages convolutional layers to generate spatial maps representing potential crack locations. Notably, DeepCrack produces these multi-scale predictions from distinct levels within the model. These predictions, each capturing details at varying levels of granularity, are then strategically combined to create the final, highly refined segmentation result. This multi-scale approach allows DeepCrack to effectively capture both fine-grained crack details and broader contextual information, leading to improved segmentation accuracy compared to models that use only a single prediction scale. However, the increased depth of the network can lead to higher computational costs and potential overfitting. The use of a GNN in our proposed approach could help mitigate these issues by providing a more efficient way to capture complex patterns without significantly increasing the depth of the network.

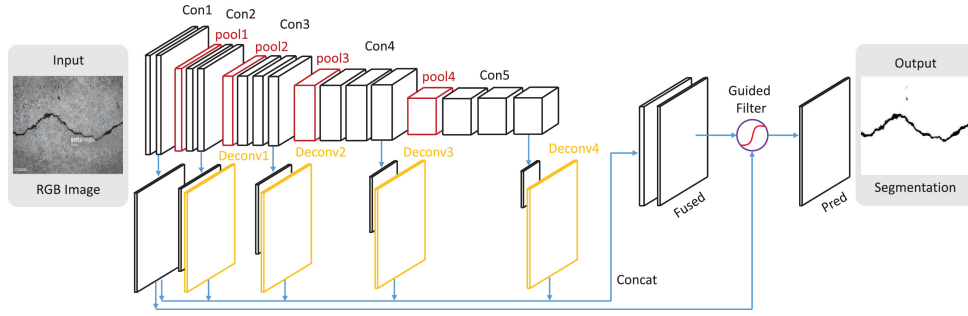


Figure 2.11: DeepCrack Architecture [8]

PSPNet [9] tackles object scale variations with a pyramid pooling module (see Fig 2.12). This allows PSPNet to capture cracks at different scales, which is crucial as cracks can vary greatly in size. However, the pyramid pooling module can sometimes struggle to accurately capture the scale variations in road crack images, leading to less precise segmentation. The use of a GNN in our proposed approach could help to address this issue by providing a more flexible and adaptive way to capture scale variations.

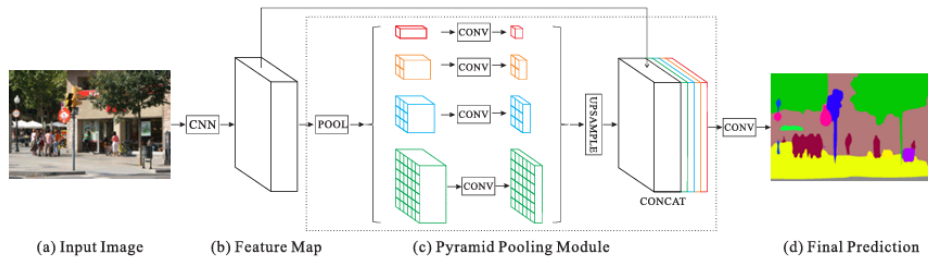


Figure 2.12: PSPNet Architecture [9]

FPHBN [10] fuses a feature pyramid with a hybrid boosting strategy for enhanced precision (see Fig 2.13). This allows FPHBN to capture cracks at different scales and orientations, which is crucial for accurate crack detection. However, the increased complexity of these models can potentially impact computational efficiency. The use of a GNN in our proposed approach could help mitigate these issues by providing a more efficient way to capture complex patterns without significantly increasing the complexity of the model.

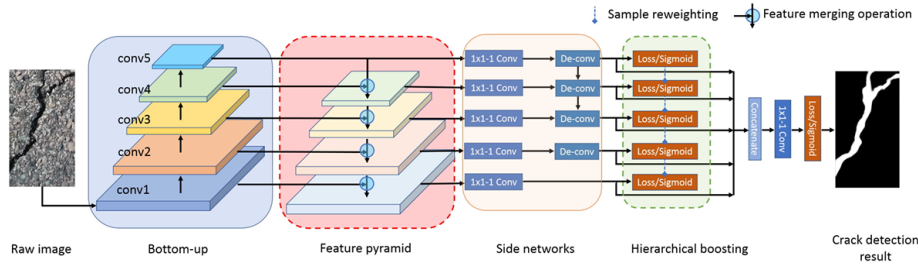


Figure 2.13: FPHBN Architecture [10]

In summary, while deep learning-based techniques have significantly advanced the field of road crack detection, they are not without their limitations. These include challenges in prioritizing the most relevant features, handling complex object scales, and dealing with noise sensitivity and imprecise boundaries. As we move forward, it becomes increasingly clear that hybrid approaches, which combine the strengths of different architectures, may hold the key to addressing these challenges. In the next section, we will explore such hybrid approaches, with a particular focus on the potential of GNNs to improve the performance of U-Net in the task of crack detection.

2.2.5 Hybrid Approaches

Researchers have been led to investigate hybrid approaches that combine the strengths of CNNs with other techniques in the ongoing pursuit of improved accuracy and robustness in crack detection. For instance, [138] integrated traditional image processing techniques with CNNs for pre-processing steps and post-processing refinement, achieving improved crack detection performance.

Attention mechanisms, inspired by human visual attention, have gained popularity in recent architectures. They help models focus on the most relevant image regions, a feature particularly useful in crack detection. Studies like [139–142] have demonstrated the effectiveness of attention mechanisms in road crack detection.

Transformer-based architectures, initially developed for natural language processing tasks, are now being explored in the domain of computer vision. Their versatility and adaptability are evident in a wide range of tasks, notably crack detection [143–146].

2.3 Summary

In the related work section, the discussion started with the importance of road crack detection and the factors that contribute to their formation. The evolution of detection methods was then traced, beginning with manual inspections, which, while straightforward, can be time-consuming, labor-intensive, and potentially miss some cracks. To improve efficiency and accuracy, the use of image processing techniques such as thresholding-based techniques, edge detection-based techniques, and feature extraction-based techniques was discussed. The exploration then transitioned to machine learning methods, both supervised and unsupervised, which have become popular because of their capacity to learn from data and provide accurate predictions. Deep learning methods, which use complex networks to learn from large volumes of data, were also explored. Attention mechanisms and transformer-based approaches were also discussed for their ability to focus on specific features and handle long-range dependencies in data. Hybrid methods, which combine traditional, machine learning, transformer, and attention-based methods to leverage the strengths of all, were examined. In essence, this section presents a thorough exploration of the development of road crack detection, from manual methods to advanced machine learning methods.

CHAPTER 3

Graph Neural Networks

Graph Neural Networks (GNNs) have gained significant attention in recent years due to their ability to effectively process and analyze data structured as graphs. Unlike traditional neural networks that operate on regular grid-like data, GNNs are designed to handle data represented as nodes and edges, capturing the intricate relationships and dependencies between entities. This unique capability makes GNNs particularly powerful for a wide range of applications, from social network analysis to molecular biology, where the connections between elements play a crucial role. In this section, we will explore the fundamental concepts of GNNs, their core operations, and various types of GNN architectures that have been developed to address different challenges in graph-based data analysis.

3.1 Introduction to Graph Neural Networks

Amidst these advancements, GNNs [50] have emerged as a powerful tool for processing data structured as graphs. They are designed to model the relationships between nodes within a graph, making them particularly suited to tasks where the connections between elements are important. In a GNN, entities are represented by nodes, while the relationships between those

entities are represented by edges. The GNN processes the graph in a way that considers both the features of the entities and the structure of the graph.

Graph G is defined by nodes (vertices) and edges, represented mathematically as:

$$G = (V, E) \quad (3.1)$$

where V signifies the set of nodes (vertices) and E signifies the set of edges. The adjacencies between nodes are captured by an adjacency matrix A , defined as:

$$A_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ have an edge} \\ 0, & \text{otherwise} \end{cases} \quad (3.2)$$

Graph Convolution: The core operation in many GNNs is graph convolution, which allows them to aggregate details from neighboring nodes in a way that respects the graph structure. This is achieved by adjusting the feature vector of each node using a weighted average that incorporates both its own features and those of its neighboring nodes. The mathematical representation of the graph convolution operation for node i in the subsequent layer $l + 1$ expressed as:

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ij}} (\mathbf{W}^{(l)} \mathbf{h}_j^{(l)} + \mathbf{b}^{(l)}) \right) \quad (3.3)$$

where:

- $\mathbf{h}_i^{(l+1)}$ is the updated feature vector for node i at layer $l + 1$.
- σ is an activation function (e.g., ReLU).

- $\mathcal{N}(i)$ is the set of neighbors of node i .
- c_{ij} is a normalization constant (often the square root of the product of the degrees of nodes i and j).
- $\mathbf{W}^{(l)}$ is a weight matrix for layer l .
- $\mathbf{h}_j^{(l)}$ is the feature vector of node j at layer l .
- $\mathbf{b}^{(l)}$ is a bias vector for layer l .

3.1.1 Types of Graph Neural Networks

There are different types of GNNs, each with its own strengths and weaknesses. Some common types include:

- **GraphSAGE:** GraphSAGE [147] is a versatile framework designed for learning node embeddings, particularly in large graphs. Its key innovation lies in the sampling and aggregation strategy:
 1. **Sampling:** For each node i , a fixed-size set of neighbors is sampled from its neighborhood $\mathcal{N}(i)$. This makes the computation feasible for large graphs.
 2. **Aggregation:** The characteristics of the sampled neighbors are combined using an aggregation function (e.g., mean, max-pooling, LSTM) to obtain a single representative vector for the neighborhood:

$$\mathbf{h}_{\mathcal{N}(i)}^{(l)} = \text{AGGREGATE}(\{\mathbf{h}_j^{(l)} | j \in \text{sampled neighbors of } i\}) \quad (3.4)$$

3. **Update:** The node's feature vector is updated based on its own features and the aggregated neighborhood features:

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\mathbf{W}^{(l)} [\mathbf{h}_i^{(l)} || \mathbf{h}_{\mathcal{N}(i)}^{(l)}] \right) \quad (3.5)$$

- **Graph Attention Network (GAT) :** GATs [148] introduce attention mechanisms into the message-passing process, enabling nodes to weigh the importance of different neighbors. This is achieved by learning attention coefficients:

$$\alpha_{ij}^{(l)} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} || \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}]))}{\sum_{k \in \mathcal{N}(i)} \exp(\text{LeakyReLU}(\mathbf{a}^T [\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} || \mathbf{W}^{(l)} \mathbf{h}_k^{(l)}]))} \quad (3.6)$$

The updated feature vector is then computed as:

$$\mathbf{H}^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l)} \right) \quad (3.7)$$

- **Graph Convolutional Network (GCN):** GCNs [149] have emerged as a prominent approach for learning from graph-structured data. Their power comes from their efficiency in extracting localized features from the graph using a convolution-like operation. This is achieved via message passing, a procedure in which nodes gather information from their neighbors and subsequently adjust their features.

The process of message-passing within GCN can be formally expressed as:

$$\mathbf{H}^{(l+1)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}) \quad (3.8)$$

where:

- $\mathbf{H}^{(l)}$ represents matrix containing the features at layer l .
- $\mathbf{W}^{(l)}$ represents adaptable weight matrix at layer l .
- $\hat{\mathbf{A}}$ represents a normalized adjacency matrix, incorporated to handle scaling and self-loop challenges:

$$\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \quad (3.9)$$

- $\mathbf{A}$ is the adjacency matrix.
- $\mathbf{D}$ is the degree matrix.
- σ is an activation function (e.g., ReLU).

By stacking multiple GCN layers, the model can capture increasingly complex relationships and dependencies between nodes. This empowers GCNs to handle a variety of tasks, encompassing node classification, graph classification, and link prediction.

GCNs have demonstrated their effectiveness in various graph-based tasks, including:

1. **Node Classification:** Assigning labels to individual nodes based on their attributes and connections within the graph. This has applications in community detection, protein function prediction, and spam detection.
2. **Link Prediction:** Estimating the probability of connections between nodes, is a crucial task for predicting future interactions in evolving networks, such as recommendation systems or drug-target interactions.
3. **Graph Classification:** This involves giving labels to whole graphs according to their structure and features. It's useful in areas like classifying chemical compounds and detecting malware.
4. **Clustering/Community Detection:** This involves grouping nodes or graphs that are similar based on their features. It's useful for finding communities within functional units in biological networks, social networks, and unusual patterns across diverse systems.
5. **Graph Embedding:** This is about showing graphs or nodes in a space with fewer dimensions, but still keeping their structure and feature information. We'll look more into this topic in the next section.

The success of GNNs in computer vision, such as scene understanding, object detection, and segmentation [150–153], highlights their ability to model intricate spatial relationships. This makes them well-suited for problems where understanding structure is crucial. Moreover, in the field of medical image segmentation [154–158], GNNs have demonstrated significant potential. Their ability to represent complex anatomical structures leads to improved accuracy and robustness in delineating organs, tumors, and other relevant features. These successes in both computer vision and medical domains underscore the potential of GNNs for road crack detection, a task that similarly relies on understanding complex spatial patterns.

3.1.2 Graph Neural Network Based Crack Detection

In the context of crack detection, representing a crack as a graph provides a powerful framework for analyzing structural relationships and connectivity patterns. This approach leverages the strengths of Graph Neural Networks (GNNs) to enhance the accuracy and robustness of crack detection methods.

Nodes and Edges in Crack Modeling

Nodes in the graph can represent distinct regions within an image, such as pixels or superpixels, while edges signify the spatial proximity or similarity between these regions. This graph representation allows the model to capture the intricate relationships between different parts of the crack.

Message Passing and Connectivity Patterns

Through the process of message passing, GNNs can learn to reason about the connectivity patterns within the graph. This involves aggregating information from neighboring nodes and updating node representations iteratively. Yang et al. [159] demonstrate the effectiveness of this approach in their work on road graph extraction, where GNNs analyze structural relationships within images. This enables better distinction between true cracks, which exhibit strong structural relationships, and isolated noise or interferences.

Enhancing Crack Continuity

GNNs can enhance crack continuity by considering long-range structural dependencies. This is particularly useful for analyzing potentially fragmented cracks, where traditional methods might struggle. By incorporating information from distant nodes, the model can maintain a coherent representation of the crack structure.

Building upon these GNN advantages, Chen et al. [160] propose a crack segmentation framework that incorporates a graph network branch. This allows the model to explicitly learn structural crack patterns, improving segmentation accuracy. The framework leverages the ability of GNNs to capture complex relationships within the graph, leading to more precise crack detection.

Similar advancements have been made in GNN-based crack detection using point cloud data. For instance, Feng et al. [161] focus on identifying fine-grained cracks through GNN-based feature analysis. Point cloud data, which represents the surface geometry of objects, can be effectively modeled as a graph, where nodes represent points in the cloud and edges represent their spatial relationships. GNNs can then analyze these relationships to detect cracks with high precision.

3.2 Summary

Graph Neural Networks offer a promising new direction for road crack detection. Their ability to explicitly model the relational structure of cracks and the context in which they appear has the potential to overcome limitations faced by purely convolutional architectures. While research into GNNs specifically for crack detection might be in its early stages, their foundational strengths and successes in related domains make them a highly compelling area for further exploration.

CHAPTER 4

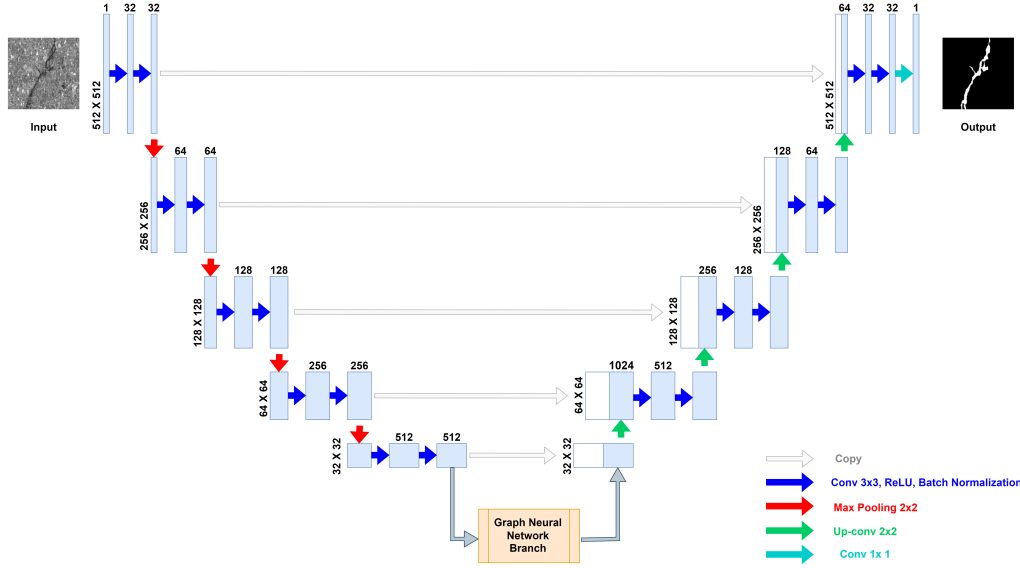
Methodology

This chapter details the research methods employed in this study to address the critical issue of pavement crack detection. Given the significant implications of road network integrity on safety, efficiency, and economic activity, the development of a reliable and automated road crack detection system is of paramount importance. This research leverages the power of modern deep learning approaches, specifically GNNs, to optimize pavement maintenance strategies, thereby enhancing road lifespan and contributing to a more sustainable transportation infrastructure. The following sections will detail the specific steps and techniques used in the development of this system.

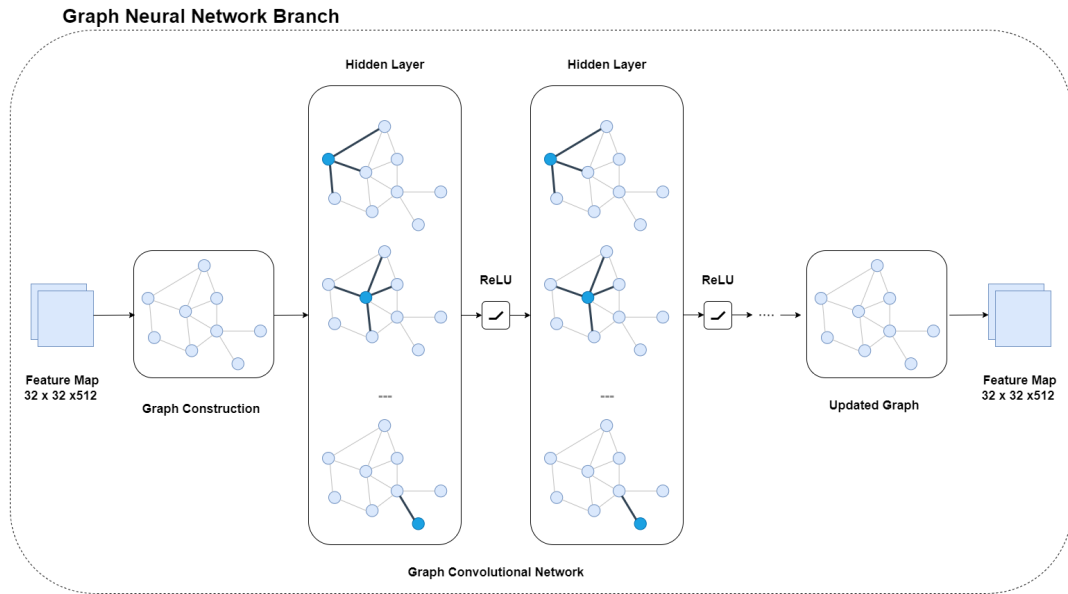
4.1 CrackGNet Architecture

The proposed CrackGNet architecture is a hybrid model that synergistically integrates the capabilities of CNNs and GNNs. The architecture is composed of two primary components: a U-Net encoder for initial feature extraction from the input image, and a GCN for refining these features by analyzing the spatial relationships and connectivity between potential cracks.

The U-Net encoder, a type of CNN, is the first component of the architecture. The input



(a) Illustration of the proposed CrackGNet architecture.



(b) Graph Neural Network Branch (detailed).

Figure 4.1: Illustration of the proposed CrackGNet architecture with a detailed view of the Graph Neural Network branch.

image undergoes a series of convolutional layers, with each layer followed by a pooling layer. These layers gradually decrease the input's spatial size while amplifying the complexity of information, extracting a multi-level representation of features from the image. The convolutional layers employ adaptable filters to identify local patterns like textures and edges within the input, whereas the pooling layers decrease the input's dimensions, enhancing the model's resilience to variations.

From the series of feature maps generated by the U-Net encoder, we select one and transform it into a graph representation. In this graph, each node represents a pixel in the feature map, and edges are established based on a distance metric. This transformation enables us to harness the capabilities of GNNs to process the image data while considering the spatial relationships between pixels.

The graph representation is then processed by the GCN, the second component of the architecture. The GCN improves the features of each node by incorporating the features of its neighbors. This is achieved through a series of graph convolution operations, which combines information from a node's local neighborhood. The result is a set of refined features that incorporate both the original features of each pixel and the context provided by its neighboring pixels.

The refined features are then converted back into a feature map, which is fed into the U-Net decoder. The U-Net decoder is a mirror image of the encoder, composed of a series of convolutional and upsampling layers that gradually expand the spatial dimensions while reducing the depth. The decoder combines the refined features from GCN with the high-level contextual features captured by the U-Net encoder to generate a precise crack segmentation map.

Finally, a 1×1 convolution is applied to the output of the U-Net decoder to produce the final crack segmentation map. This operation collapses the feature map into a single channel, producing a single-channel output where each pixel represents the model's prediction for

its corresponding location in the original input image. The values in this map indicate the likelihood of each pixel belonging to a crack, providing a detailed and precise segmentation of the cracks in the pavement.

4.2 Feature Extraction using U-Net Encoder

The initial stage of our proposed CrackGNet architecture involves feature extraction from the input image using a U-Net encoder. The input to this process is typically a grayscale image of size $512 \times 512 \times 1$. The U-Net encoder, a type of CNN specifically designed for image segmentation tasks, processes this image through a sequence of convolutional layers, pooling layers, and activation functions.

The primary goal of this stage is to derive hierarchical features from the given image. The convolutional layers are responsible for identifying patterns and textures of increasing complexity. These layers employ adaptable filters on the input data, identifying local patterns like edges and textures. As the image progresses through the network, the convolutional layers learn to recognize more complex patterns, which are combinations of the simpler patterns learned by the earlier layers.

Conversely, the pooling layers gradually reduce the level of detail captured in the feature maps. This downsampling operation serves to capture broader contextual information, making the model more robust to handle variations in the input data. It also diminishes the computational complexity of the network, as the following layers operate on smaller input sizes.

Our U-Net encoder employs a standard architecture, progressively increasing the channel depths as the image features are downsampled. The channel depths are as follows: 32, 64, 128, 256, and 512 (see Fig 4.1a - left to Graph Neural Network branch). This gradual escalation in channels allows the model to learn representations of cracks at various scales, from fine details to larger patterns.

Upon processing by the U-Net encoder, a series of feature maps with varying spatial resolutions and channel depths are generated. We strategically select the $32 \times 32 \times 512$ feature map for subsequent graph processing. This feature map, with a height $h = 32$, width $w = 32$, and $c = 512$ channels, offers several advantages. It maintains a sufficient level of spatial detail for accurate crack localization, provides a rich feature representation with 512 channels, and strikes a balance with computational efficiency.

We experimented with feature maps from neighboring layers in the U-Net encoder. Earlier layers with higher spatial resolutions, while beneficial for precise localization, lead to significantly larger graphs. For example, a 64×64 feature map would generate $N = h \times w = 4096$ nodes. This increased graph size impacts both the memory requirements and the computational cost of the subsequent GNN operations, hindering efficiency. Conversely, later layers with smaller resolutions proved too coarse, sacrificing the spatial detail essential for accurate crack identification. Therefore, the $32 \times 32 \times 512$ feature map was chosen as a compromise between spatial detail, feature richness, and computational efficiency.

4.3 Graph Construction

Road cracks inherently exhibit intricate spatial relationships and connectivity patterns that traditional convolutional neural networks struggle to fully capture. To address this challenge, we meticulously transform the selected $32 \times 32 \times 512$ feature map into a graph representation, empowering us to explicitly model these structural dependencies (as shown in Algorithm 1).

In our graph, each node directly represents a single pixel within the feature map, inheriting the rich 512 features extracted by the U-Net encoder. These features encapsulate essential crack characteristics derived from the image.

Several approaches exist for constructing graphs in the realm of crack detection. For instance, graphs can be built leveraging domain-specific knowledge about how crack patterns appear in images. Alternatively, similarity or distance-based methods [162] define edges based

on calculated similarity in features or spatial positions. Another common method is K-Nearest Neighbours (KNN) [163], where each node connects to its ‘K’ most similar neighbors.

We adopted a distance-based approach, similar to the work of Chen et al. [160]. This choice aligns with their demonstrated success in modeling crack patterns. We prioritize modeling both local crack continuity and potential branching by establishing edges based on a distance metric ‘d’. Through rigorous experimentation with different ‘d’ values, we identified their inherent limitations. For instance, smaller ‘d’ values definitively miss longer-range crack connections, restricting the ability to model complex crack patterns. Conversely, larger ‘d’ values lead to overly dense graphs that increase computational demands during GNN operations. This trade-off is illustrated by the relationship between ‘d’, the number of nodes ‘N’, and the resulting edges ‘E’: $E = f(d, N)$. Examples include $d = 1$ yielding $E_{max} = 4N$, while $d = 2$ results in $5N \leq E \leq 12N$ (see Fig 4.2), and a larger $d = 3$ could generate up to $20N$ edges. Our comprehensive experiments demonstrated that $d = 2$ strikes the optimal balance between capturing the essential structure of cracks and maintaining computational efficiency for our specific crack patterns. However, we acknowledge that other values of ‘d’ might be optimal in scenarios where cracks exhibit substantially different patterns.

Algorithm 1 Pseudocode for constructing a graph from encoder output

```
1: Function CreateGraph(encoder_output, d)
2:   reshaped_output  $\leftarrow$  Reshape(encoder_output)
3:   graph  $\leftarrow$  InitializeEmptyGraph()
4:   For each node in reshaped_output
5:     features  $\leftarrow$  GetFeatures(node)
6:     neighbors  $\leftarrow$  GetNeighbors(node, d)
7:     For each neighbor in neighbors
8:       AddEdge(graph, node, neighbor)
9:     End For
10:    SetNodeFeatures(graph, node, features)
11:  End For
12:  FormatGraph(graph)
13:  Return graph
14: End Function
```

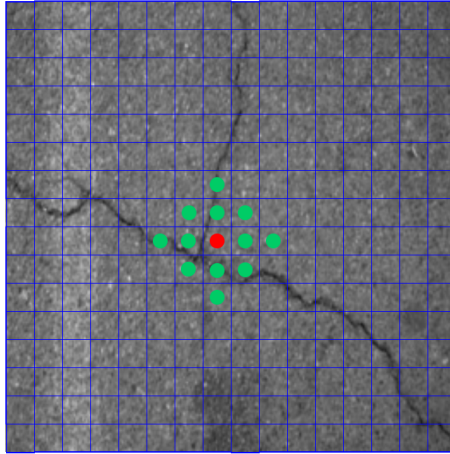


Figure 4.2: A small input image with the corresponding graph overlaid (showing node and edge formation based on the distance metric)

By strategically modeling crack structures in graph form, we enable the GCN to reason about the spatial context and connectivity of potential crack segments. This improves the precision and reliability of crack detection, as the model can now utilize the structural information represented in the graph to make more knowledgeable predictions. This graph-based approach represents a significant advancement over traditional CNN-based methods, providing a more thorough understanding of the spatial relationships between pixels and leading to more accurate and reliable crack detection.

4.4 Feature Refinement using Graph Convolutional Network (GCN)

The constructed graph, encoding both spatial relationships and rich feature representations of potential cracks, is strategically processed by a GCN [149]. Unlike traditional CNNs designed for grid-like image data, GCNs are uniquely equipped to learn from the intricate connectivity patterns within graphs. We carefully considered alternatives like GAT [148] and GraphSAGE [147]. However, GCN provides the optimal balance for our task due to the

relatively homogeneous nature of our crack patterns and the need for computational efficiency [164].

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad (4.1)$$

where:

- $\mathbf{H}^{(l)}$ is a matrix where each row corresponds to the feature vector of a node at the layer l .
- $\mathbf{W}^{(l)}$ is the learnable weight matrix at layer l . It transforms the node features into a new feature space.
- $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ is the adjacency matrix with added self-loops (the identity matrix $\mathbf{I}$ is added to allow each node to include its own features in the aggregation).
- $\tilde{\mathbf{D}}$ represents degree matrix corresponding to $\tilde{\mathbf{A}}$, a diagonal matrix where $\tilde{D}_{ii}$ is the degree of node i in $\tilde{\mathbf{A}}$.
- $\mathbf{b}^{(l)}$ is the bias term associated with layer l .
- σ is an activation function (e.g., ReLU).

GCN layers perform specialized convolutions as described in Equation 4.1 that fundamentally differ from their traditional counterparts. Instead of operating on regular pixel grids, GCN directly leverages the graph structure. Each node dynamically gathers feature information from its connected neighbors and modifies its own representation based on this aggregated neighborhood information (see Fig. 4.1b). This process captures the spatial context surrounding each potential crack segment.

4.5 Feature Map Reconstruction and U-Net Decoder

Fusion

In the subsequent stages of our model, we focus on the intricate process of integrating the enriched feature representations obtained from the GCN back into the U-Net architecture. This integration is crucial for translating the graph-based learning of the GCN into a form that can be used for pixel-wise crack segmentation. The process is divided into two main steps: the reconstruction of the feature map from the GCN output, and the fusion of this refined feature map with the U-Net decoder. The following sections explain the details of these steps and their significance in our model.

Reconstruction

After the GCN has processed the graph, enriching the crack representations with spatial and relational context, we must transform these refined features back into a format compatible with the U-Net architecture. This involves meticulously reshaping the graph's node features to reconstruct a feature map with the original dimensionality $32 \times 32 \times 512$. Specifically, we perform a reverse operation of the initial reshaping process, mapping each node's features to its corresponding spatial location in the feature map. This reconstructed feature map retains the spatial structure of the original image, but with features that have been enhanced by the GCN's graph-based learning process.

Fusion

The refined feature map now serves as input to the U-Net decoder. The decoder strategically combines this GCN-refined information with the high-level contextual features captured earlier by the U-Net encoder using skip connections. While techniques like concatenation or element-wise summation exist for fusion [165], they might not be ideal. Concatenation

increases dimensionality and computational burden, while element-wise summation might not effectively balance the importance of different features. We leverage skip connections within the U-Net decoder. This fusion process is performed at each layer of the decoder, progressively integrating the refined features with the encoder's multi-scale representations (see Fig. 4.1a - right to Graph Neural Network Branch). This fusion offers several key advantages for accurate crack segmentation:

- **Enhanced Localization:** The GCN-refined features provide detailed, spatially-aware crack representations, improving the model's ability to precisely localize crack boundaries. This is particularly important for accurately delineating the extent of each crack, which is crucial for subsequent repair planning.
- **Robustness to Noise:** By incorporating connectivity information learned by the GCN, the model becomes more resilient to noise and visual artifacts that might otherwise resemble cracks. This helps to reduce false positives, ensuring the model's predictions are reliable and actionable.
- **Contextual Awareness:** The fusion with high-level encoder features ensures the model considers broader image context, preventing misclassification of crack-like structures that are inconsistent with the overall scene. This helps to further reduce false positives, particularly in complex or cluttered environments.

Decoder Operation

The decoder leverages skip connections to merge the refined features with the matching features of the same spatial resolution from the encoder path. This fusion allows the decoder to utilize both the detailed crack representations refined by the GCN and the broader contextual information captured across multiple scales by the U-Net encoder. The decoder then employs a series of convolutional and upsampling operations to progressively increase the spatial

resolution of the feature maps, ultimately producing a segmentation map with the same dimensions as the original image. The specific upsampling method used in our decoder is transposed convolution, which allows for learning-based upsampling that can better capture the complex patterns of crack structures.

4.6 Final Convolution

The last step in our model is to convert the high-dimensional feature maps output by the U-Net decoder into a final crack segmentation map. This is achieved through the application of a 1×1 convolution, a common technique in semantic segmentation models for channel reduction and pixel-wise prediction. The role and benefits of this operation are as follows:

- **Channel Reduction:** The 1×1 convolution serves as a mechanism to decrease the depth of the feature maps. The U-Net decoder produces a feature map with multiple channels, each corresponding to a detailed representation of the features found in the input image. However, for the final segmentation map, we need a 2D map where every pixel represents a binary class label ('crack' or 'non-crack' in the case of binary segmentation). The 1×1 convolution achieves this by applying a filter to each spatial location across all channels, effectively decreasing the depth of the feature map to match the desired number of output classes.
- **Pixel-wise Prediction:** The 1×1 convolution operation also enables the model to make pixel-wise predictions. Each pixel in the output map corresponds to a location in the original input image, and its value represents the model's prediction for that location. In the case of binary segmentation, this could be a probability score indicating the likelihood of the pixel belonging to the crack class. This pixel-wise prediction capability allows the model to generate a detailed segmentation map that precisely delineates the boundaries of detected cracks.

4.7 Loss Function

The loss function is essential for the training process of deep learning models. It quantifies the divergence between the model's predictions and the ground truth, providing a measure that the optimization algorithm seeks to minimize. For our crack detection task, we employ the Dice Loss function [166], a prevalent choice for segmentation tasks due to its robustness to class imbalance and direct alignment with evaluation metrics. The Dice Loss function is calculated using the following formula:

$$\text{Dice Loss} = 1 - \frac{2 * |X \cap Y|}{|X| + |Y|} \quad (4.2)$$

where X represents the predicted segmentation map and Y the ground truth.

The reasons for choosing Dice Loss for our model are:

- **Handling Class Imbalance:** In crack detection tasks, the datasets often exhibit significant class imbalance, with the pixels representing cracks forming a small percentage of the overall image. This imbalance can lead to a model that is prejudiced towards predicting the majority class, resulting in poor performance of the minority (crack) class. Dice Loss, by focusing on the overlap between the predicted and actual crack regions, is less sensitive to this class imbalance compared to alternatives like cross-entropy loss.
- **Alignment with Evaluation Metrics:** The Dice coefficient, alternatively referred to as the Sørensen–Dice coefficient, is a commonly used metric for assessing the performance of image segmentation models. It measures the convergence between the predicted and actual areas, offering a score between 0 (no overlap) and 1 (perfect overlap). By using Dice Loss, we ensure that our model is directly optimizing for the same identical metric that is used to assess its performance, leading to more consistent and interpretable results.

While other loss functions such as focal loss [167] or cross-entropy [168] are also used in segmentation tasks, we found that Dice Loss provides the most robust and accurate results for our specific task. Its focus on region overlap makes it particularly suited to tasks like crack detection, where accurately delineating the boundaries of the crack regions is of paramount importance.

4.8 Summary

The proposed CrackGNet model integrates CNN and GNN for efficient crack detection. It employs a U-Net encoder for initial feature extraction from the input image, generating a series of feature maps. One of these maps is transformed into a graph representation, which is processed by a GCN to refine the features based on spatial relationships and connectivity patterns of potential cracks. The refined features are transformed back into a feature map and fed into the U-Net decoder, which combines these refined features with the high-level contextual features captured by the U-Net encoder to generate a precise crack segmentation map. The graph is constructed from the selected feature map, where each node represents a pixel, and edges are created based on a distance metric. After the GCN processing, the refined features are transformed back into a feature map, combined with high-level contextual information from the U-Net encoder, and a 1×1 convolution is applied to the U-Net decoder's output to produce the final crack segmentation map. The Dice Loss function, a suitable choice for datasets with significant class imbalance, is used to train this model.

CHAPTER 5

Experimental Results

5.1 Experimental Setup

To comprehensively assess the performance of our proposed CrackGNet method, we conducted an extensive experimental study. This section outlines the experimental setup details, including software framework, hardware, model architecture, hyperparameter settings, and training procedure.

Software Framework

- **PyTorch:** Our model was implemented using PyTorch [169], a popular deep-learning framework known for its flexibility and user-friendly interface. PyTorch’s dynamic computational graph and extensive library of modules and functions facilitated the development of our hybrid CNN-GCN architecture. Its support for GPU acceleration also ensured efficient computation, which was crucial given the complexity of our model and the size of our datasets.

Optimization

- **Optimizer:** We selected the AdamW optimizer [170] for its adaptive learning rate adjustments and decoupled weight decay. These features help in faster convergence and potential generalization improvements, making it a suitable choice for our complex model.
- **Hyperparameters:** The hyperparameters were carefully tuned through a series of experiments and validation. Comprehensive details regarding these hyperparameters are available in Table 5.1.

Hyperparameter	Value/Description
Input image size	512×512 (balances computational efficiency and detail)
Learning rate	1e-3 (balances training speed and model performance)
Loss weights	1.0 for fused layer and side-output (equal contribution to final loss)
Momentum	0.9 (accelerates optimizer and dampens oscillations)
Weight decay	1e-4 (prevents overfitting)
Number of epochs	100 (sufficient for convergence and prevents overfitting)

Table 5.1: Model hyperparameters and their values

Hardware and Training

- **Computational Resources:** Our experiments were conducted on Google Colab, leveraging the computational power of a Tesla T4 GPU. This allowed us to train our model on large datasets within a reasonable time frame.
- **Training Strategy:** We employed a performance-based model saving strategy, preserving only the model weights that yielded improvements on a validation set. This approach ensures that the final model represents the best-performing configuration throughout the training process, thereby maximizing its performance on unseen data.

5.2 Datasets

A comprehensive evaluation of crack detection algorithms requires the use of diverse and challenging datasets that accurately reflect real-world conditions. Our study utilized the following publicly accessible datasets for training and evaluating the performance of our CrackGNet model:

Crack Forest Dataset (CFD)

- **Origin and Purpose:** CFD [171] was specifically designed for the evaluation of crack detection algorithms on concrete pavement surfaces. It offers an accurate depiction of the complexities encountered in real-world crack detection, such as diverse lighting conditions, various crack orientations, and varying crack widths.
- **Composition:** The dataset includes 118 images, each having a resolution of 480×320 pixels. It presents challenges due to the presence of background noise elements, including shadows, stains, and variations in pavement texture. These factors make it a rigorous test for any crack detection algorithm.
- **Annotations:** Each image is meticulously annotated with ground-truth crack contours, providing a basis for quantitative performance evaluation. These annotations allow for precise evaluation of the model's capacity to detect and delineate cracks.

The CFD dataset served as a crucial benchmark for evaluating CrackGNet's ability to handle complex pavement textures and distinguish true cracks from visual interferences. It was used in both the training and validation stages of our study.

DeepCrack

- **Origin and Purpose:** DeepCrack dataset [8] aims to advance research in automated crack detection by providing a publicly accessible dataset focused on concrete surfaces.

Its purpose was to facilitate the development and evaluation of deep learning algorithms designed for crack detection.

- **Composition:** DeepCrack contains total 537 images, each measuring 544×384 pixels. The dataset encompasses cracks of varying widths, lengths, and orientations, representing the diverse nature of cracks found in real-world scenarios. This diversity makes it an ideal dataset for training robust and generalizable models.
- **Annotations:** Manual labeling of all images ensures precise pixel-level ground truth for accurate evaluation of segmentation performance. These detailed annotations enable a rigorous assessment of the model's performance.

DeepCrack was instrumental in both the training and validation processes of CrackGNet. Its diversity of crack patterns and real-world complexities helped refine the model's ability to generalize to a broad range of crack detection challenges. It was used extensively in the training phase to ensure a robust learning process.

CRACK500

- **Origin and Purpose:** CRACK500 dataset [10] is a large-scale dataset explicitly designed to address the challenges of pavement crack detection. It was collected using a mobile phone to reflect practical image acquisition conditions, making it a highly realistic dataset for training and evaluating crack detection models.
- **Composition:** CRACK500 offers a substantial collection of 3368 images, providing a richer and more varied representation of crack patterns compared to smaller datasets. The images include various crack types (linear, alligator, block) and complex pavement backgrounds, enhancing model robustness. The extensive image collection ensures that the model encounters a wide range of crack scenarios, promoting better generalization.

- **Annotations:** Pixel-level annotations are provided for each image, enabling precise evaluation. These annotations are essential for evaluating the model’s ability to precisely find and delineate cracks.
- **Data Split:** The dataset is divided into a training segment comprising 1896 images and a testing segment comprising 1124 images, enabling a systematic evaluation process. This split assures that the model is assessed on unseen data, providing a reliable measure of its performance.

CRACK500 was instrumental in both the training and rigorous testing of CrackGNet. The dataset’s scale and complexity were essential for pushing the model’s generalization capabilities. It was used extensively in both the training and testing phases of our study.

5.3 Baseline Models

To provide a comprehensive baseline for evaluating the performance of our CrackGNet method, we selected a range of crack detection models based on deep learning as baselines. These models represent key architectural advancements and offer diverse approaches to the problem of identifying cracks in images.

- **U-Net[5]:** U-Net serves as a fundamental pillar in the field of image segmentation, including crack detection. Its encoder-decoder structure systematically extracts hierarchical features at different scales, while skip connections enable precise localization during decoding. U-Net excels in its ability to effectively learn and combine both low-level and high-level features for image segmentation but can sometimes struggle with the complexities of crack detection, such as handling noise, intricate crack structures, and variations in lighting.
- **SegNet[6]:** SegNet specifically aims to improve boundary delineation in segmentation tasks. It shares a similar encoder-decoder architecture with U-Net but uniquely stores

max-pooling indices during the encoding phase, which are then reused during upsampling for a sharper recovery of object boundaries. SegNet's emphasis on boundary preservation can be advantageous for crack detection where accurate crack localization is crucial, however, its primary reliance on convolutional operations might limit its ability to model complex spatial relationships inherent in crack patterns.

- **Attention U-Net [7]:** Attention U-Net is an extension of the U-Net architecture by introducing attention gates. These gates enable the model to dynamically prioritize the most relevant image regions at different stages of processing, suppressing noise and irrelevant background elements. The attention mechanism enhances segmentation accuracy, particularly in images with complex backgrounds or where cracks are obscured, but might still be insufficient to fully capture the intricate connectivity and structural patterns of cracks, particularly in challenging scenarios.
- **DeepCrack[8]:** DeepCrack differentiates itself with a significantly deeper encoder-decoder architecture compared to U-Net. This increased depth allows the model to extract a wider range of hierarchical features, including subtle and intricate crack details. DeepCrack demonstrates an enhanced ability to detect fine-grained cracks that might be missed by shallower architectures, but the reliance on a deep convolutional network might lead to overfitting or challenges in generalization, especially with limited datasets, and can increase computational costs.
- **FPHBN[10]:** FPHBN seeks to improve segmentation precision by combining a feature pyramid and a hybrid boosting strategy. The feature pyramid within its encoder-decoder structure captures multi-scale crack representations, while the boosting approach refines segmentation predictions. FPHBN's design makes it effective in handling cracks of varying sizes and complexities, but the complexity introduced by the feature pyramid and boosting strategy can lead to longer training times and increased computational

overhead.

- **PSPNet [9]:** PSPNet tackles the issue of object scale variations by incorporating a pyramid pooling module into a typical encoder-decoder architecture. This module captures global context information at different resolutions, making PSPNet more robust in detecting cracks of diverse sizes within a single image. However, the increased architectural complexity can lead to potentially higher computational costs.

5.4 Evaluation Metrics

In order to rigorously assess the performance of our CrackGNet method, we employed a comprehensive suite of evaluation metrics commonly used in image segmentation tasks. These metrics provide insights into different aspects of the model's crack detection capabilities, enabling a multi-faceted analysis.

Precision

- **Definition:** Precision quantifies the percentage of correctly identified crack pixels (true positives, TP) out of the total pixels the model classified as cracks (true positives + false positives, FP).
- **Interpretation:** A high precision score indicates that the model minimizes false alarms, ensuring that the majority of its crack predictions are accurate.
- **Importance:** Precision is essential to ensure the reliability of crack identification, especially when used for automated maintenance decision-making. High precision reduces the risk of unnecessary interventions based on false detections.

Recall

- **Definition:** Recall evaluates how effectively a model can identify all actual crack pixels within an image, comparing the number of correctly detected crack pixels (true positives TP) against the total number of crack pixels present (true positives plus any missed cracks, or false negatives FN).
- **Interpretation:** A high recall score signifies that the model effectively minimizes missed detections and is sensitive to even faint or subtle cracks.
- **Importance:** Recall is vital for detecting even the faintest cracks, crucial for early intervention, and preventing potential structural damage. High recall ensures that all potential threats are identified, even if some false positives are included.

F-Score

- **Definition:** The F-score is a measure of a model's performance that balances precision and recall. It's calculated as their harmonic mean, offering a single value to represent the model's overall effectiveness.
- **Interpretation:** A high F-score suggests that the model skillfully avoids both false alarms and missed detections, achieving a good balance between precision and recall.
- **Importance:** The F-Score provides a comprehensive assessment of the model's performance, considering both the reliability of crack detection and the completeness of detected cracks. It is particularly useful when there is no clear preference between precision and recall.

Intersection over Union (IoU)

- **Definition:** IoU provides a straightforward evaluation of the accuracy of the predicted crack segmentation's location. It quantifies the overlap between the two areas as a

proportion of their total combined area. Values range from 0 (no overlap) to 1 (perfect overlap).

- **Interpretation:** A high IoU score demonstrates the model’s proficiency in precisely delineating crack boundaries.
- **Importance:** IoU focuses on the spatial accuracy of predicted crack boundaries, which is crucial for determining the exact location and extent of damage. High IoU ensures that the model not only identifies cracks but also accurately segments them, which is essential for subsequent repair planning.

Mathematical Formulation

For clarity, we provide the mathematical formulas for these metrics:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (5.1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5.2)$$

$$\text{F-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5.3)$$

$$\text{IoU} = \frac{|\text{Predicted Mask} \cap \text{Ground Truth Mask}|}{|\text{Predicted Mask} \cup \text{Ground Truth Mask}|} \quad (5.4)$$

5.5 Experimental Results

We conducted a rigorous evaluation of our CrackGNet method and compared its performance against established baselines across the CFD, DeepCrack, and CRACK500 datasets. Our analysis includes both qualitative assessment through visualizations and quantitative evaluation using the metrics outlined in the previous section.

Figure 5.1 demonstrates the visual quality of crack segmentations generated by CrackGNet and the comparison methods. It highlights CrackGNet’s ability to accurately delineate crack boundaries, suppress noise, and handle complex crack patterns. In parallel, Figure 5.2 visually represents the Training-Validation loss and IoU curves for these methods. During the training process, we selected the best model at the point where the loss value was at its minimum.

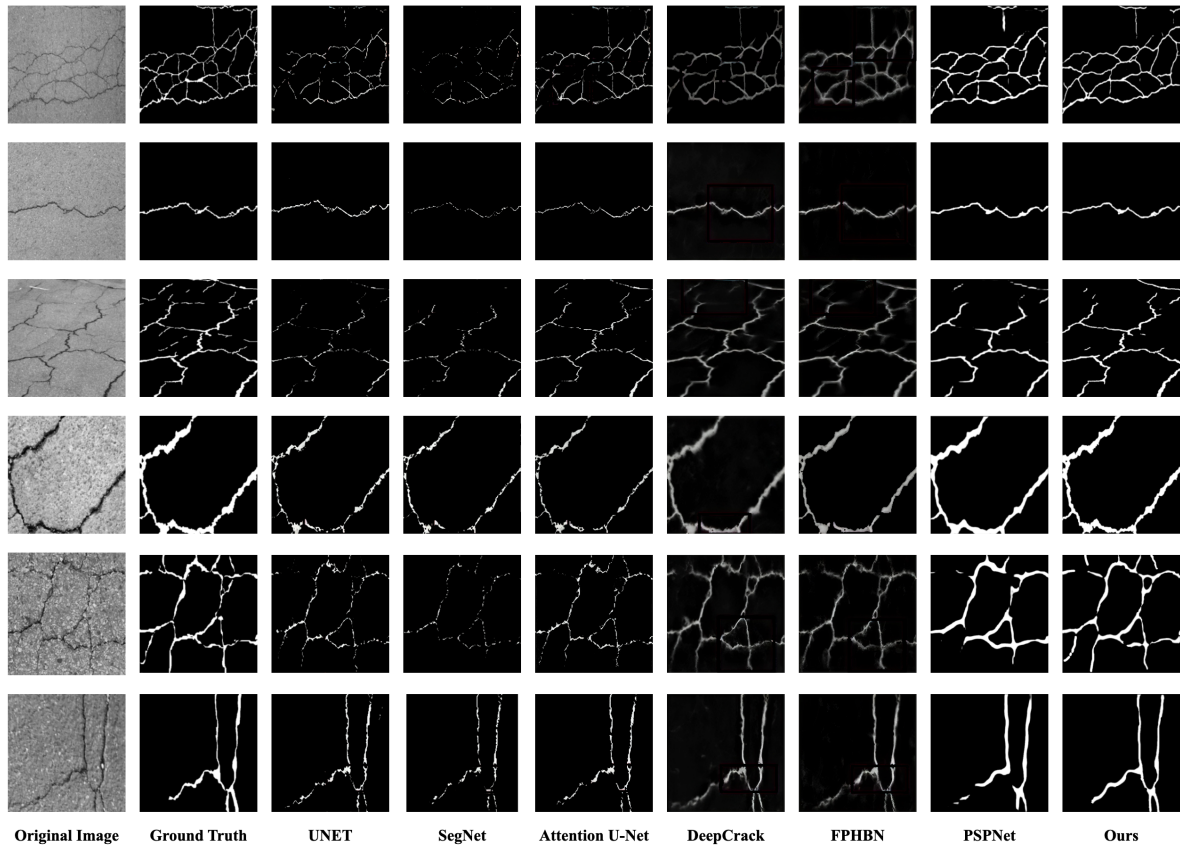


Figure 5.1: Visual assessment of results from various methods on selected samples from the CFD, DeepCrack, and Crack500 datasets. The methods compared include U-Net, SegNet, Attention U-Net, DeepCrack, FPHBN, PSPNet, and the method proposed in this work.

Tables 5.2, 5.3, and 5.4 present detailed numerical results for the CFD, DeepCrack, and CRACK500 datasets, respectively. CrackGNet consistently achieves superior performance across the majority of metrics, particularly in Precision and Recall.

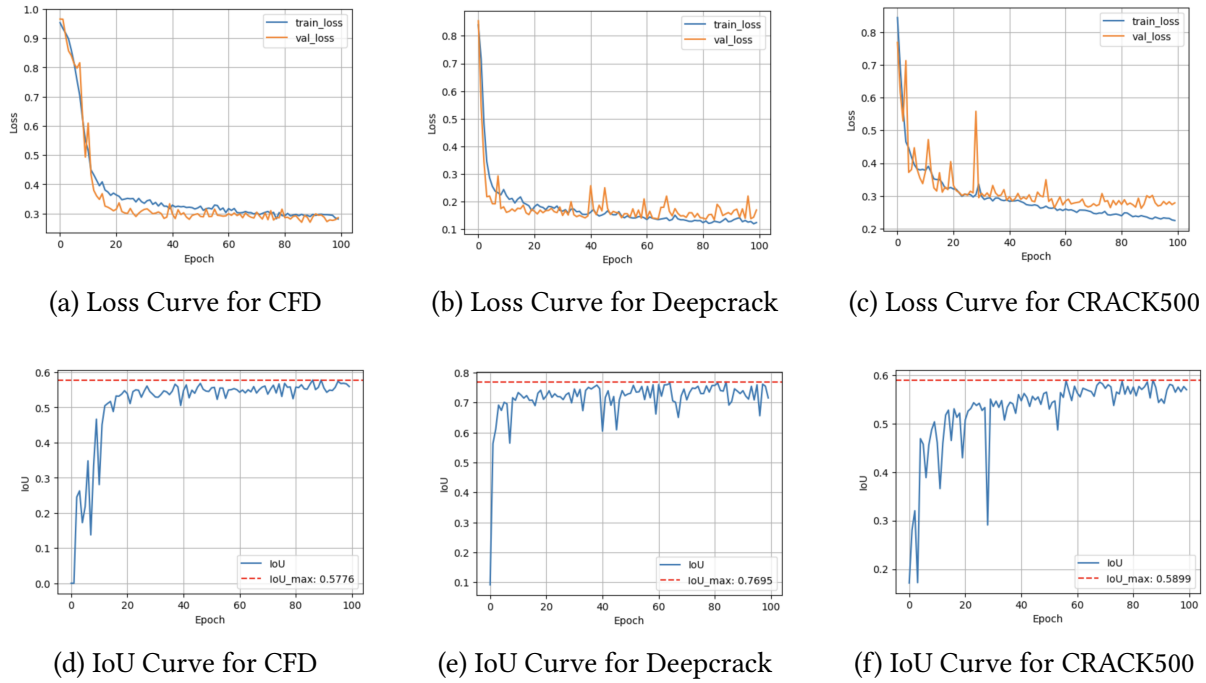


Figure 5.2: Loss and IoU Curves for all datasets

5.5.1 Evaluation of Results on CFD

The CFD dataset is notably challenging due to the existence of noise elements and complex pavement textures. Despite these difficulties, our proposed CrackGNet method demonstrated its robustness and effectiveness. As shown in Table 5.2, it achieved a Precision of 0.7096, a Recall of 0.7608, and a resulting F1-Score of 0.7313. These results significantly surpass those of the next best-performing method, PSPNet, which achieved an F1-Score of 0.6468. This substantial improvement underscores the ability of CrackGNet to accurately discriminate true cracks from visually similar background elements. The GCN component of CrackGNet, which models spatial relationships, likely plays a crucial role in handling the noisy textures characteristic of the CFD dataset (see Fig 5.3). This performance on the CFD dataset suggests that CrackGNet is highly effective in real-world road crack detection scenarios where image quality may be less than ideal.

Table 5.2: CFD Dataset Results

Method	Precision	Recall	F1-Score
U-Net [5]	0.4824	0.6146	0.5404
SegNet [6]	0.5434	0.2974	0.3834
Attention U-Net [7]	0.5230	0.5922	0.5554
DeepCrack [8]	0.5770	0.6733	0.6214
FPHBN [10]	0.5628	0.6781	0.6150
PSPNet [9]	0.6040	0.7024	0.6468
Ours	0.7096	0.7608	0.7313

Table 5.3: Deepcrack Dataset Results

Method	Precision	Recall	F1-Score
U-Net [5]	0.8206	0.8301	0.8253
SegNet [6]	0.8412	0.7369	0.7850
Attention U-Net [7]	0.8532	0.8148	0.8335
DeepCrack [8]	0.8529	0.8478	0.8504
FPHBN [10]	0.8387	0.8072	0.8224
PSPNet [9]	0.8424	0.8070	0.8187
Ours	0.8618	0.8811	0.8665

5.5.2 Evaluation of Results on DeepCrack

The DeepCrack dataset provides a diverse range of crack patterns and scales, offering insights into CrackGNet’s ability to handle a variety of crack types. As detailed in Table 5.3, our method attained a Precision of 0.8618, a Recall of 0.8811, and an F1-Score of 0.8665. This performance surpasses the established DeepCrack model, which achieved an F1-Score of 0.8504. The integration of the GCN component in CrackGNet likely enables it to better capture intricate crack structures compared to architectures that rely solely on convolutional layers (see Fig 5.4). This leads to improved results on the DeepCrack dataset.

5.5.3 Evaluation of Results on CRACK500

The large-scale CRACK500 dataset offers a rigorous test of a model’s ability to infer across different scenarios. As shown in Table 5.4, our CrackGNet method excelled in this test, achieving a Precision of 0.7741, a Recall of 0.7136, and an F1-Score of 0.7374. This performance

Table 5.4: CRACK500 Dataset Results

Method	Precision	Recall	F1-Score
U-Net[5]	0.5985	0.6017	0.6001
SegNet[6]	0.7002	0.4068	0.5146
Attention U-Net [7]	0.6165	0.6047	0.6106
DeepCrack [8]	0.6437	0.6535	0.6486
FPHBN [10]	0.6004	0.5816	0.5908
PSPNet [9]	0.7495	0.7004	0.7135
Ours	0.7741	0.7136	0.7374

outperforms the next-best method, PSPNet, which achieved an F1-Score of 0.7135. The consistent superiority of CrackGNet across multiple metrics, particularly on a challenging dataset like CRACK500, highlights the robustness of our GCN-enhanced architecture for complex crack detection tasks (see Fig 5.5).

5.5.4 Intersection over Union (IoU) Results

In addition to the previously discussed metrics, we also evaluated the IoU for our model, providing a further measure of segmentation accuracy. As detailed in Table 5.5, CrackGNet achieved IoU values of 0.5899, 0.7695, and 0.5776 on the CFD, DeepCrack, and CRACK500 datasets, respectively. These results underscore the ability of our model to produce segmentations that closely match the ground truth annotations. The notably high IoU score on the DeepCrack dataset further indicates that our method is particularly adept at accurately delineating crack boundaries, even when faced with a diverse array of crack patterns.

Dataset	IoU
CFD	0.5776
DeepCrack	0.7695
CRACK500	0.5899

Table 5.5: IoU values for the three datasets.

5.6 Discussion of Overall Results

The results of this research demonstrate significant advancements in the domain of road crack detection, aligning with the contributions outlined in the introduction. The hybrid deep learning architecture, which integrates U-Net’s feature extraction capabilities with the structural modeling power of Graph Neural Networks (GNNs), has proven to be highly effective. This innovative approach excels in detecting fine cracks, which are typically challenging for conventional methods, as seen in Figure 5.3 and 5.5. The enhanced detection capability is crucial for timely maintenance interventions, thereby contributing to improved road safety and maintenance efficiency. Additionally, the hybrid model exhibits remarkable robustness and reliability in real-world scenarios, effectively adapting to varying lighting conditions, complex textures, and the presence of visually similar non-crack features, as illustrated in Figure 5.4. One of the standout features of the proposed approach is its ability to model long-range dependencies within crack patterns using GNNs, significantly enhancing the detection of complex crack structures. The model effectively identifies and maps long-range cracks, as shown in Figure 5.3. Furthermore, the integration of GNNs in the proposed model has shown potential for improved computational efficiency, reducing computational overhead and making the proposed model a promising solution for real-time or resource-constrained applications.

5.7 Discussion of Model Components

The CrackGNet architecture integrates multiple components that work in synergy to achieve superior crack detection performance. In this section, we delve into the role of each component and its contribution to the overall effectiveness of our model.

U-Net Component

The foundational U-Net component serves as a powerful feature extractor. Its encoder-decoder structure, incorporating skip connections, allows for the progressive extraction of hierarchical features at multiple scales. This capability is essential for accurately identifying crack structures that exhibit variations in width, length, and orientation. Furthermore, the skip connections in the decoder facilitate precise crack localization by fusing fine-grained details with higher-level contextual information. The strong performance of our CrackGNet model across all datasets, particularly the high Precision, Recall, and F1 scores, provides evidence of the U-Net component's essential role in the overall architecture.

GNN Component

The incorporation of the GCN component is a key innovation in CrackGNet. GCNs excel at modeling the spatial relationships and connectivity patterns inherent in crack structures. While traditional CNNs in the U-Net can struggle to fully capture these complex dependencies, the GCN component explicitly reasons about such relationships. This enhanced ability to discriminate between true cracks and visually confounding background elements likely contributes significantly to the superior performance of CrackGNet, as demonstrated by the high Recall and F1 scores across the evaluated datasets.

Hybrid Approach

The strategic fusion of the U-Net and GCN components forms the core of CrackGNet's hybrid design. This combination capitalizes on the complementary strengths of each component:

- **U-Net:** Robust extraction of crack-related features and precise localization.
- **GCN:** Enhanced modeling of crack patterns and spatial context.

The consistent performance gains across multiple datasets underscore the success of this

hybrid approach. While a formal ablation study was not conducted for this work, we hypothesize that removing either component would likely diminish the model's overall performance. This suggests that both the U-Net and GCN components play a significant role in CrackGNet's effectiveness for complex crack detection tasks.

5.8 Summary

The experimental setup for the proposed CrackGNet model was implemented on PyTorch, using an encoder-decoder structure with batch normalization and ReLU activations. The model was optimized using AdamW, with specific hyperparameters, and trained for 100 epochs. The performance was evaluated on three datasets: CFD, DeepCrack, and CRACK500, each offering a unique set of images and crack types. The model was compared against six baseline models, namely U-Net, SegNet, Attention U-Net, DeepCrack, FPHBN, and PSPNet, each with their unique architectural and functional strengths. The evaluation metrics used were Precision, Recall, F-score, and IoU, calculated based on the comparison of detected results with the hand-labeled ground truth. The model demonstrated superior performance across all datasets, indicating its effectiveness in distinguishing cracks from visually similar features and its robustness to variations in crack types and image quality.

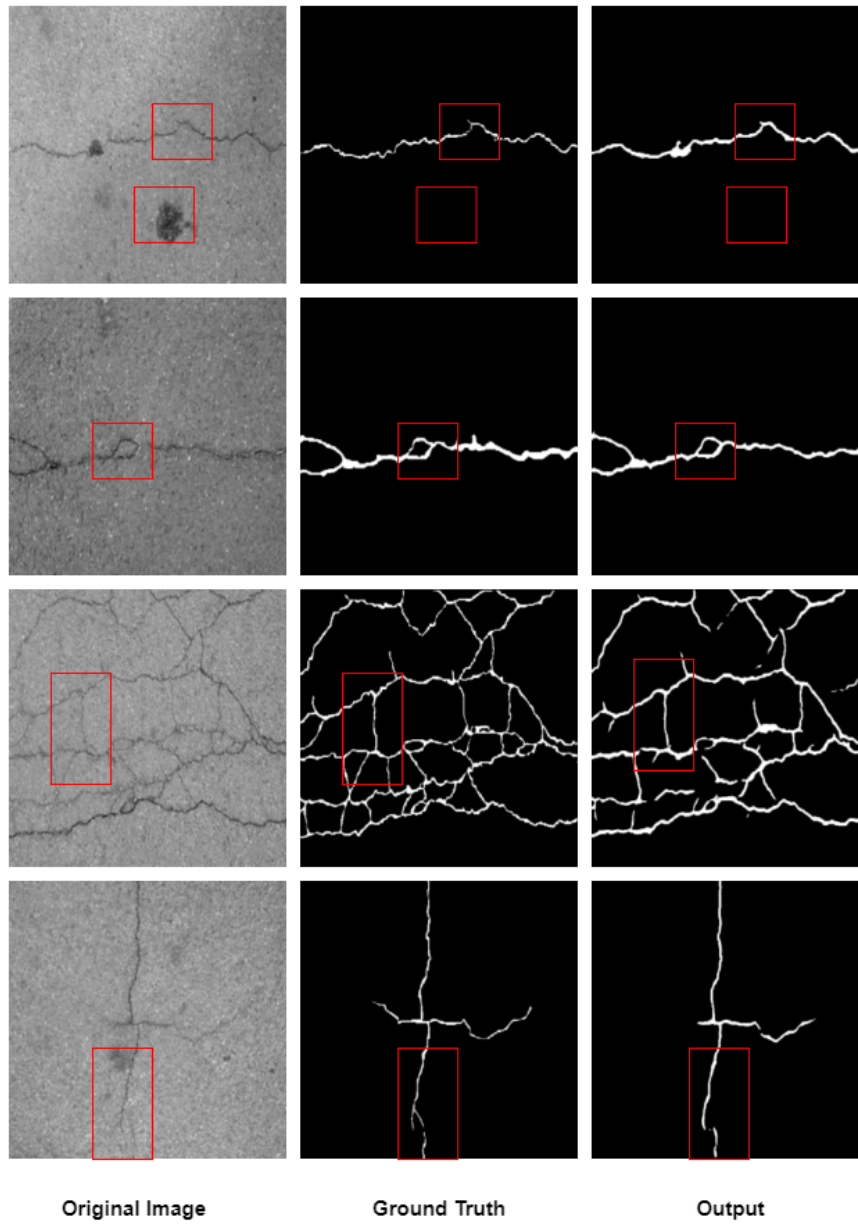


Figure 5.3: Examples of crack detection from CFD

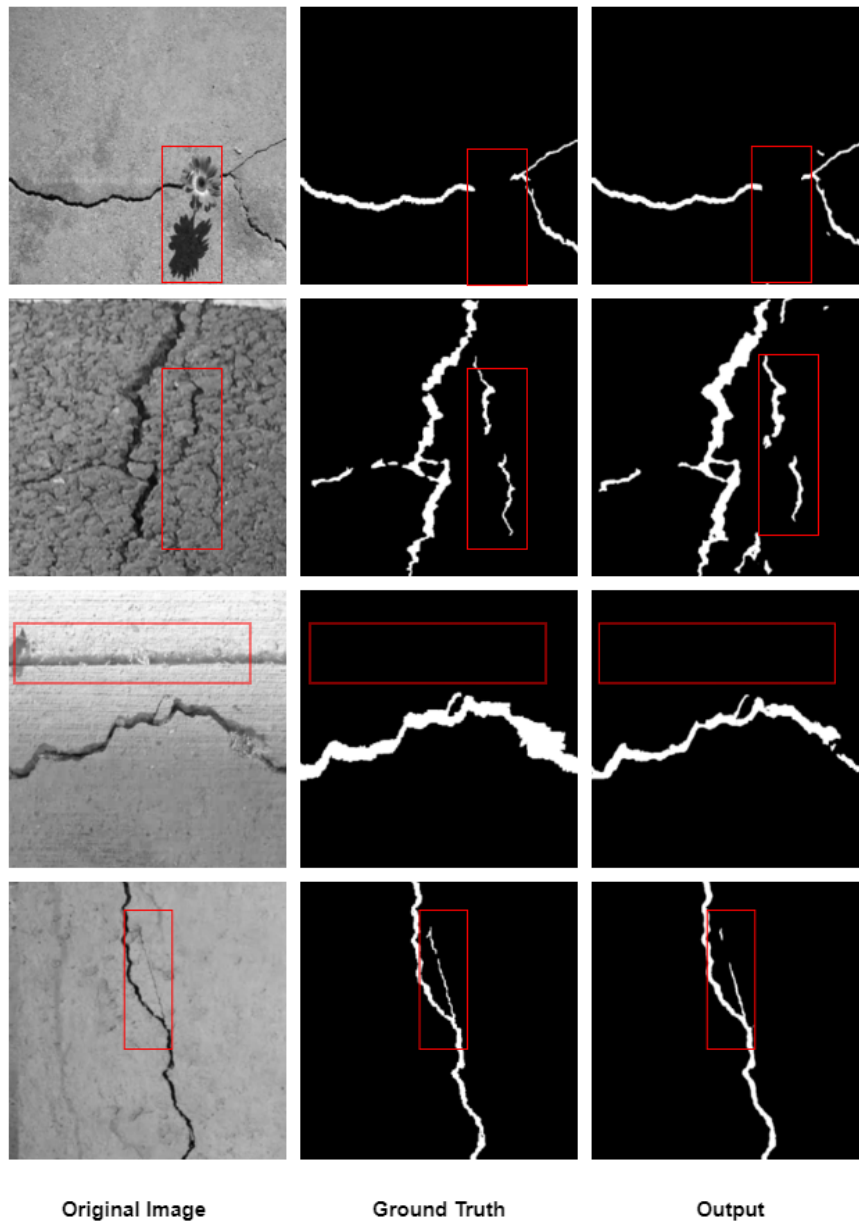


Figure 5.4: Examples of crack detection from DeepCrack

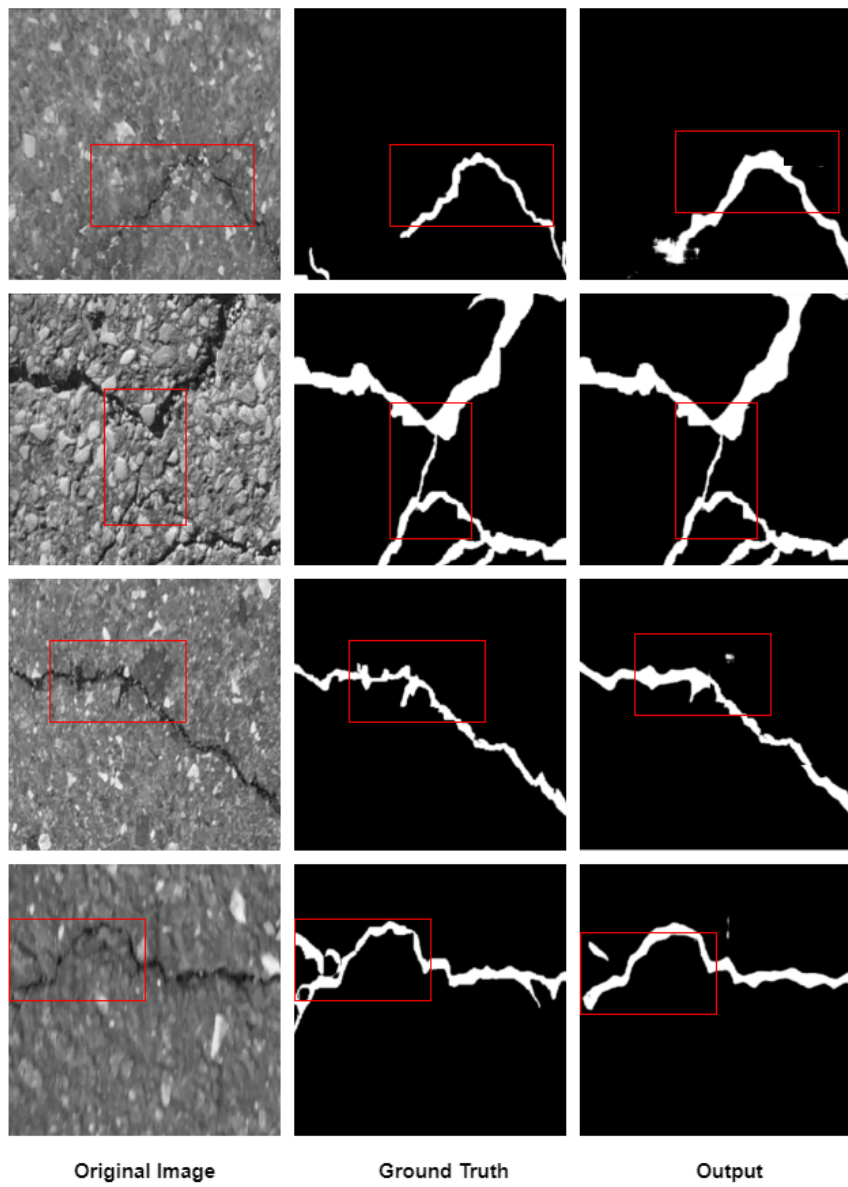


Figure 5.5: Examples of crack detection from CRACK500

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

This thesis presented CrackGNet, a hybrid deep-learning architecture designed for the challenging task of automated road crack detection. Our proposed method strategically combines the strengths of a U-Net encoder-decoder for feature extraction with a GCN to explicitly model the complex spatial relationships and connectivity patterns inherent in cracks. Extensive experimentation on the CFD, DeepCrack, and CRACK500 datasets demonstrated CrackGNet's superiority over established deep-learning baselines across a comprehensive suite of evaluation metrics. This performance advantage underscores the effectiveness of our hybrid approach and the following key contributions:

- **GCN Integration:** By incorporating a GCN component, CrackGNet effectively learns to model the intricate structure and spatial dependencies within crack patterns. This enhanced capability is particularly beneficial for discriminating between true cracks and visually similar background elements in complex pavement images.
- **Hybrid Design:** The fusion of convolutional feature extraction (U-Net) and graph-

based reasoning (GCN) within CrackGNet creates a powerful framework for crack detection. This integration allows the model to leverage the strengths of both the U-Net, known for its precise localization, and the GCN, renowned for its ability to comprehend spatial relationships.

The results achieved with CrackGNet indicate its strong potential for practical application in automated pavement inspection and maintenance systems.

6.2 Future Work

While the CrackGNet framework presented in this thesis demonstrates significant advancements in the field of automated crack detection, several promising avenues for future research remain to be explored.

First, we aim to further optimize the computational efficiency of our GNN architecture. By investigating lighter-weight models or pruning techniques, we can streamline the inference process, enabling real-time crack detection in practical settings.

Second, we intend to delve deeper into alternative graph construction strategies. This involves exploring different approaches to node and edge feature representation, as well as incorporating domain-specific knowledge to ensure a comprehensive and accurate representation of crack dependencies within the graph structure.

Third, to enhance the generalizability and robustness of our model, we plan to significantly expand our training dataset. This expansion will encompass a wider range of real-world crack images from diverse sources, as well as the potential incorporation of synthetically generated crack examples with varying characteristics.

Fourth, building upon the features learned by our GNN model, we will extend our research to develop automated tools for assessing the severity of detected cracks. This will involve defining appropriate severity metrics, establishing thresholds, and potentially incorporating

additional information (e.g., crack dimensions, and material properties) to provide actionable insights for maintenance and repair decisions.

Finally, we envision integrating our crack detection and severity assessment system with robotic inspection platforms. This would enable autonomous, large-scale inspection of infrastructure, streamlining maintenance processes and potentially reducing costs.

Bibliography

- [1] H. Khawaja and A. Tanveer, “Review of low-temperature crack (ltc) developments in asphalt pavements,” 2018. → [p1], [p8]
- [2] T. I. Limited, “What are the types of cracking in roads?,” September 2023. → [p1], [p16], [p17], [p18]
- [3] H. Fu, D. Meng, W. Li, and Y. Wang, “Bridge crack semantic segmentation based on improved deeplabv3+,” *Journal of Marine Science and Engineering*, vol. 9, no. 6, p. 671, 2021. → [p1], [p23]
- [4] Y.-A. Hsieh and Y. J. Tsai, “Machine learning for crack detection: Review and model performance comparison,” *Journal of Computing in Civil Engineering*, vol. 34, no. 5, p. 04020038, 2020. → [p1], [p32]
- [5] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241, Springer, 2015. → [p1], [p11], [p34], [p35], [p67], [p74], [p75]

- [6] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
→ [p1], [p35], [p36], [p67], [p74], [p75]
- [7] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, *et al.*, “Attention u-net: Learning where to look for the pancreas,” *arXiv preprint arXiv:1804.03999*, 2018.
→ [p1], [p36], [p37], [p68], [p74], [p75]
- [8] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, “Deepcrack: A deep hierarchical feature learning architecture for crack segmentation,” *Neurocomputing*, vol. 338, pp. 139–153, 2019.
→ [p1], [p37], [p38], [p65], [p68], [p74], [p75]
- [9] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2881–2890, 2017.
→ [p1], [p38], [p69], [p74], [p75]
- [10] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling, “Feature pyramid and hierarchical boosting network for pavement crack detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1525–1535, 2019.
→ [p1], [p11], [p38], [p39], [p66], [p68], [p74], [p75]
- [11] J. Lekshmipathy, N. M. Samuel, and S. Velayudhan, “Vibration vs. vision: Best approach for automated pavement distress detection,” *International Journal of Pavement Research and Technology*, vol. 13, no. 4, pp. 402–410, 2020.
→ [p7]
- [12] TRIP, “Meeting america’s transportation needs with a reliable, safe well-maintained national highway network,” in *America’s Interstate Highway System at 65*, October 2021.
→ [p7]

- [13] S. Tarawneh and M. Sarireh, "Causes of cracks and deterioration of pavement on highways in Jordan from contractors' perspective," *Civil and Environmental Research*, vol. 3, no. 10, pp. 16–26, 2013. → [p7], [p8]
- [14] Y. Yao, S.-T. E. Tung, and B. Glisic, "Crack detection and characterization techniques—an overview," *Structural Control and Health Monitoring*, vol. 21, no. 12, pp. 1387–1413, 2014. → [p7], [p8]
- [15] H. Nhat-Duc, Q.-L. Nguyen, and V.-D. Tran, "Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network," *Automation in Construction*, vol. 94, pp. 203–213, 2018. → [p7], [p8], [p9], [p11], [p21], [p22]
- [16] Y. O. Ouma and M. Hahn, "Wavelet-morphology based detection of incipient linear cracks in asphalt pavements from rgb camera imagery and classification using circular radon transform," *Advanced Engineering Informatics*, vol. 30, no. 3, pp. 481–499, 2016. → [p7], [p8], [p21]
- [17] E. Zaloshnja and T. R. Miller, "Cost of crashes related to road conditions, united states, 2006," in *Annals of Advances in Automotive Medicine/Annual Scientific Conference*, vol. 53, p. 141, Association for the Advancement of Automotive Medicine, 2009. → [p8]
- [18] M. Gavilán, D. Balcones, O. Marcos, D. F. Llorca, M. A. Sotelo, I. Parra, M. Ocaña, P. Aliseda, P. Yarza, and A. Amírola, "Adaptive road crack detection system by pavement classification," *Sensors*, vol. 11, no. 10, pp. 9628–9657, 2011. → [p8], [p11], [p21], [p22]
- [19] Z. Zhu, S. German, and I. Brilakis, "Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation," *Automation in Construction*, vol. 20, no. 7, pp. 874–883, 2011. → [p9]

- [20] I. Al-Saadi, H. Wang, X. Chen, P. Lu, and A. Jasim, "Multi-objective optimization of pavement preservation strategy considering agency cost and environmental impact," *International Journal of Sustainable Transportation*, vol. 15, no. 11, pp. 826–836, 2021.
→ [p9]
- [21] H. Wang, I. Al-Saadi, P. Lu, and A. Jasim, "Quantifying greenhouse gas emission of asphalt pavement preservation at construction and use stages using life-cycle assessment," *International Journal of Sustainable Transportation*, vol. 14, no. 1, pp. 25–34, 2020.
→ [p9]
- [22] D. of Transport (United Kingdom), "Road safety statistics," May 2022.
→ [p9]
- [23] R. S. Lim, H. M. La, Z. Shan, and W. Sheng, "Developing a crack inspection robot for bridge maintenance," in *2011 IEEE International Conference on Robotics and Automation*, pp. 6288–6293, IEEE, 2011.
→ [p9]
- [24] P. Prasanna, K. J. Dana, N. Gucunski, B. B. Basily, H. M. La, R. S. Lim, and H. Parvardeh, "Automated crack detection on concrete bridges," *IEEE Transactions on automation science and engineering*, vol. 13, no. 2, pp. 591–599, 2014.
→ [p9], [p29]
- [25] W. Wang, M. Wang, H. Li, H. Zhao, K. Wang, C. He, J. Wang, S. Zheng, and J. Chen, "Pavement crack image acquisition methods and crack extraction algorithms: A review," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 6, no. 6, pp. 535–556, 2019.
→ [p9], [p11], [p21], [p22]
- [26] H. Zakeri, F. M. Nejad, A. Fahimifar, A. D. Torshizi, and M. F. Zarandi, "A multi-stage expert system for classification of pavement cracking," in *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*, pp. 1125–1130, IEEE, 2013.
→ [p10]

- [27] Q. Li, Q. Zou, D. Zhang, and Q. Mao, "Fosa: F* seed-growing approach for crack-line detection from pavement images," *Image and Vision Computing*, vol. 29, no. 12, pp. 861–872, 2011. → [p11]
- [28] Q. Li and X. Liu, "Novel approach to pavement image segmentation based on neighboring difference histogram method," in *2008 Congress on Image and Signal Processing*, vol. 2, pp. 792–796, IEEE, 2008. → [p]
- [29] F. Liu, G. Xu, Y. Yang, X. Niu, and Y. Pan, "Novel approach to pavement cracking automatic detection based on segment extending," in *2008 International Symposium on Knowledge Acquisition and Modeling*, pp. 610–614, IEEE, 2008. → [p]
- [30] A. Akagic, E. Buza, S. Omanovic, and A. Karabegovic, "Pavement crack detection using otsu thresholding for image segmentation," in *2018 41st international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pp. 1092–1097, IEEE, 2018. → [p25]
- [31] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement crack detection using the gabor filter," in *16th international IEEE conference on intelligent transportation systems (ITSC 2013)*, pp. 2039–2044, IEEE, 2013. → [p]
- [32] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba, "Automation of pavement surface crack detection using the continuous wavelet transform," in *2006 International Conference on Image Processing*, pp. 3037–3040, IEEE, 2006. → [p]
- [33] L. Jinhui, "Pavement crack diseases detecting by image processing algorithm," *Journal of Chang'an University*, vol. 24, no. 3, pp. 24–29, 2004. → [p27]
- [34] B. Santhi, G. Krishnamurthy, S. Siddharth, and P. Ramakrishnan, "Automatic detection of cracks in pavements using edge detection operator," *Journal of Theoretical and Applied Information Technology*, vol. 36, no. 2, pp. 199–205, 2012. → [p27]

- [35] T. Yamaguchi and S. Hashimoto, “Fast crack detection method for large-size concrete surface images using percolation-based image processing,” *Machine Vision and Applications*, vol. 21, pp. 797–809, 2010. → [p28]
- [36] K. Ehrig, J. Goebbels, D. Meinel, O. Paetsch, S. Prohaska, and V. Zobel, “Comparison of crack detection methods for analyzing damage processes in concrete with computed tomography,” in *DIR 2011-International symposium on digital industrial raidology and computed tomography (Proceedings)*, no. DGZfP-BB 128 (Poster 2), pp. 1–8, Deutsche Gesellschaft für Zerstörungsfreie Prüfung eV (DGZfP), 2011. → [p28]
- [37] R. Medina, J. Llamas, E. Zalama, and J. Gómez-García-Bermejo, “Enhanced automatic detection of road surface cracks by combining 2d/3d image processing techniques,” in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 778–782, IEEE, 2014. → [p28]
- [38] H. J. Lim and H. Sohn, “Fatigue crack detection using structural nonlinearity reflected on linear ultrasonic features,” *Journal of Applied Physics*, vol. 118, no. 24, 2015. → [p28]
- [39] W. Xu, Z. Tang, J. Zhou, and J. Ding, “Pavement crack detection based on saliency and statistical features,” in *2013 IEEE International Conference on Image Processing*, pp. 4093–4097, IEEE, 2013. → [p28], [p29]
- [40] X.-J. Yao, T.-H. Yi, C. Qu, and H.-N. Li, “Blind modal identification using limited sensors through modified sparse component analysis by time-frequency method,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 769–782, 2018. → [p29]
- [41] X. Kong and J. Li, “Vision-based fatigue crack detection of steel structures using video feature tracking,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 783–799, 2018. → [p29]

- [42] M. C. Ho, J. D. Lin, and C. F. Huang, "Automatic image recognition of pavement distress for improving pavement inspection," *GEOMATE Journal*, vol. 19, no. 71, pp. 242–249, 2020. → [p29]
- [43] J. Lin and Y. Liu, "Potholes detection based on svm in the pavement distress image," in *2010 ninth international symposium on distributed computing and applications to business, engineering and science*, pp. 544–547, IEEE, 2010. → [p29]
- [44] T. Saar and O. Talvik, "Automatic asphalt pavement crack detection and classification using neural networks," in *2010 12th Biennial Baltic Electronics Conference*, pp. 345–348, IEEE, 2010. → [p30]
- [45] N. Yusof, M. K. Osman, Z. Hussain, M. H. M. Noor, A. Ibrahim, N. Tahir, and N. Abidin, "Automated asphalt pavement crack detection and classification using deep convolution neural network," in *2019 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 215–220, IEEE, 2019. → [p]
- [46] M. R. Jahanshahi and S. F. Masri, "A novel crack detection approach for condition assessment of structures," in *Computing in Civil Engineering (2011)*, pp. 388–395, 2011. → [p30]
- [47] H. Oliveira, J. J. Caeiro, and P. L. Correia, "Improved road crack detection based on one-class parzen density estimation and entropy reduction," in *2010 IEEE International Conference on Image Processing*, pp. 2201–2204, IEEE, 2010. → [p30]
- [48] H. Oliveira and P. L. Correia, "Automatic road crack detection and characterization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 155–168, 2012. → [p28], [p30]

- [49] E. Buza, S. Omanovic, and A. Huseinovic, "Pothole detection with image processing and spectral clustering," in *Proceedings of the 2nd International Conference on Information Technology and Computer Networks*, vol. 810, p. 4853, 2013. → [p11], [p30]
- [50] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," *arXiv preprint arXiv:1810.00826*, 2018. → [p11], [p41]
- [51] M. Ameri, A. Mansourian, M. H. Khavas, M. Aliha, and M. Ayatollahi, "Cracked asphalt pavement under traffic loading—a 3d finite element analysis," *Engineering Fracture Mechanics*, vol. 78, no. 8, pp. 1817–1826, 2011. → [p19]
- [52] J. C. Pais, S. I. Amorim, and M. J. Minhoto, "Impact of traffic overload on road pavement performance," *Journal of transportation Engineering*, vol. 139, no. 9, pp. 873–879, 2013. → [p19]
- [53] N. R. Bukhari and M. Y. Aman, "Review study of identify main factor that causes the cracking and potholes on asphalt pavement in malaysia," *Recent Trends in Civil Engineering and Built Environment*, vol. 4, no. 2, pp. 338–350, 2023. → [p19]
- [54] S. Luo, T. Bai, M. Guo, Y. Wei, and W. Ma, "Impact of freeze–thaw cycles on the long-term performance of concrete pavement and related improvement measures: A review," *Materials*, vol. 15, no. 13, p. 4568, 2022. → [p19]
- [55] D. Rys, M. Jaczewski, M. Pszczola, A. Kamedulska, and B. Kamedulski, "Factors affecting low-temperature cracking of asphalt pavements: Analysis of field observations using the ordered logistic model," *International Journal of Pavement Engineering*, vol. 24, no. 2, p. 2065273, 2023. → [p19]
- [56] Y. Qiao, G. W. Flintsch, A. R. Dawson, and T. Parry, "Examining effects of climatic factors on flexible pavement performance and service life," *Transportation research record*, vol. 2349, no. 1, pp. 100–107, 2013. → [p19]

- [57] X. Lian, H. Hu, T. Li, and D. Hu, "Main geological and mining factors affecting ground cracks induced by underground coal mining in shanxi province, china," *International Journal of Coal Science & Technology*, vol. 7, no. 2, pp. 362–370, 2020. → [p20]
- [58] Q. Xie, L. Gaohu, H. Chen, C. Xu, and B. Feng, "Seismic damage to road networks subjected to earthquakes in nepal, 2015," *Earthquake Engineering and Engineering Vibration*, vol. 16, pp. 649–670, 2017. → [p20]
- [59] A. M. El-Maissi, S. A. Argyroudis, and F. M. Nazri, "Seismic vulnerability assessment methodologies for roadway assets and networks: A state-of-the-art review," *Sustainability*, vol. 13, no. 1, p. 61, 2020. → [p20]
- [60] D. P. Johnston and R. W. Surdahl, "Influence of mixture design and environmental factors on continuously reinforced concrete pavement cracking," *Transportation Research Record*, vol. 2020, no. 1, pp. 83–88, 2007. → [p20]
- [61] J. Xin, J. Pei, M. Akiyama, R. Li, J. Zhang, and L. Shao, "A study on the design method for the material composition of small particle-size asphalt mixture for controlling cracks in asphalt pavement," *Applied Sciences*, vol. 9, no. 10, p. 1988, 2019. → [p20]
- [62] E. F. Freitas, P. A. Pereira, L. P. Santos, and A. T. Papagiannakis, "Construction quality, temperature and rutting effect on top-down cracking initiation," 2005. → [p20]
- [63] S. S. Adlinge and A. Gupta, "Pavement deterioration and its causes," *International journal of innovative research and development*, vol. 2, no. 4, pp. 437–450, 2013. → [p20]
- [64] A. Dawson, *Water in road structures: movement, drainage & effects*, vol. 5. Springer Science & Business Media, 2008. → [p20]
- [65] B. Odum-Ewuakye and N. Atttoh-Okine, "Sealing system selection for jointed concrete pavements—a review," *Construction and Building Materials*, vol. 20, no. 8, pp. 591–602, 2006. → [p20]

- [66] J. France-Mensah and W. J. O'Brien, "Budget allocation models for pavement maintenance and rehabilitation: Comparative case study," *Journal of Management in Engineering*, vol. 34, no. 2, p. 05018002, 2018. → [p21]
- [67] S. Yaqoob, *Concrete pavements' repair techniques and numerical assessment of dowel bar load transfer efficiency*. PhD thesis, KTH Royal Institute of Technology, 2024. → [p21]
- [68] X. Wang, H. Zhaozheng, N. Li, and L. Qin, "Pavement crack analysis by referring to historical crack data based on multi-scale localization," *Plos one*, vol. 15, no. 8, p. e0235171, 2020. → [p22]
- [69] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced engineering informatics*, vol. 29, no. 2, pp. 196–210, 2015. → [p22]
- [70] P. Gupta and M. Dixit, "Image-based crack detection approaches: a comprehensive survey," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 40181–40229, 2022. → [p22]
- [71] N. Kheradmandi and V. Mehranfar, "A critical review and comparative study on image segmentation-based techniques for pavement crack detection," *Construction and Building Materials*, vol. 321, p. 126162, 2022. → [p22]
- [72] K. M. Chua and L. Xu, "Simple procedure for identifying pavement distresses from video images," *Journal of transportation engineering*, vol. 120, no. 3, pp. 412–431, 1994. → [p23]
- [73] W.-H. Tsai, "Moment-preserving thresholding: A new approach," *Computer vision, graphics, and image processing*, vol. 29, no. 3, pp. 377–393, 1985. → [p23]

- [74] A. M. A. Talab, Z. Huang, F. Xi, and L. HaiMing, "Detection crack in image using otsu method and multiple filtering in image processing techniques," *Optik*, vol. 127, no. 3, pp. 1030–1033, 2016. → [p23]
- [75] F. M. Abubakar, "Study of image segmentation using thresholding technique on a noisy image," *International journal of science and research (IJSR)*, vol. 2, no. 1, pp. 2319–7064, 2013. → [p24]
- [76] N. Katakam, "Pavement crack detection system through localized thresholing," Master's thesis, University of Toledo, 2009. → [p24]
- [77] H. N. Koutsopoulos and I. El Sanhoury, *Methods and algorithms for automated analysis of pavement images*. No. 1311, 1991. → [p24]
- [78] H. N. Koutsopoulos and A. Downey, "Primitive-based classification of pavement crack-ing images," *Journal of Transportation Engineering*, vol. 119, no. 3, pp. 402–418, 1993. → [p24]
- [79] H. N. Koutsopoulos, I. E. Sanhoury, and A. B. Downey, "Analysis of segmentation algorithms for pavement distress images," *Journal of transportation engineering*, vol. 119, no. 6, pp. 868–888, 1993. → [p24]
- [80] H. Oh, N. W. Garrick, and L. E. Achenie, "Segmentation algorithm using iterative clipping for processing noisy pavement images," in *Imaging Technologies: Techniques and Applications in Civil Engineering. Second International Conference Engineering Foundation; and Imaging Technologies Committee of the Technical Council on Computer Practices, American Society of Civil Engineers.*, 1998. → [p25]

- [81] C. Chen, H. Seo, Y. Zhao, B. Chen, J. Kim, Y. Choi, and M. Bang, "Automatic pavement crack detection based on image recognition," in *International Conference on Smart Infrastructure and Construction 2019 (ICSIC) Driving data-informed decision-making*, pp. 361–369, ICE Publishing, 2019. → [p25]
- [82] H. Oliveira and P. L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," in *2009 17th European Signal Processing Conference*, pp. 622–626, IEEE, 2009. → [p25], [p30]
- [83] Y.-C. Tsai, V. Kaul, and R. M. Mersereau, "Critical assessment of pavement distress segmentation methods," *Journal of transportation engineering*, vol. 136, no. 1, pp. 11–19, 2010. → [p25]
- [84] E. Salari and G. Bao, "Pavement distress detection and severity analysis," in *Image Processing: Machine Vision Applications IV*, vol. 7877, pp. 107–116, SPIE, 2011. → [p25]
- [85] E. Salari and D. Ouyang, "An image-based pavement distress detection and classification," in *2012 IEEE International Conference on Electro/Information Technology*, pp. 1–6, IEEE, 2012. → [p25]
- [86] H. Xu, Z. Li, Y. Jiang, and J. Huang, "Pavement crack detection based on opencv and improved canny operator," *Computer engineering and design*, vol. 35, no. 12, 2014. → [p25]
- [87] H. Lokeshwor, L. K. Das, and S. Goel, "Robust method for automated segmentation of frames with/without distress from road surface video clips," *Journal of Transportation Engineering*, vol. 140, no. 1, pp. 31–41, 2014. → [p25]

- [88] S. Hongxun, W. Weixing, W. Fengping, W. Linchun, and W. Zhiwei, "Pavement crack detection by ridge detection on fractional calculus and dual-thresholds," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, no. 4, pp. 19–30, 2015. → [p25]
- [89] S. Liu, "Method of shadow pavement crack extraction based on improved local threshold segmentation," *Wireless Internet Technol*, vol. 15, no. 20, pp. 112–3, 2018. → [p25]
- [90] Y. Hu, J. Wang, X. Ai, X. Zhuang, *et al.*, "An improved multithreshold segmentation algorithm based on graph cuts applicable for irregular image," *Mathematical Problems in Engineering*, vol. 2019, 2019. → [p26]
- [91] N. Safaei, O. Smadi, B. Safaei, and A. Masoud, "Efficient road crack detection based on an adaptive pixel-level segmentation algorithm," *Transportation Research Record*, vol. 2675, no. 9, pp. 370–381, 2021. → [p26]
- [92] M. V. Dharampal, "An improved type-1 fuzzy logic method for edge detection," *Int J Advance Res Comp Eng Tech*, vol. 4, pp. 1640–1644, 2015. → [p26]
- [93] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis and machine vision*. Springer, 2013. → [p26]
- [94] S. S. Al-Amri, N. Kalyankar, and S. Khamitkar, "Image segmentation by using edge detection," *International journal on computer science and engineering*, vol. 2, no. 3, pp. 804–807, 2010. → [p26]
- [95] B. Poornima, Y. Ramadevi, and T. Sridevi, "Threshold based edge detection algorithm," *International Journal of engineering and technology*, vol. 3, no. 4, p. 400, 2011. → [p26]
- [96] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of computing in civil engineering*, vol. 17, no. 4, pp. 255–263, 2003. → [p26], [p28]

- [97] A. Samant and H. Adeli, "Feature extraction for traffic incident detection using wavelet transform and linear discriminant analysis," *Computer-Aided Civil and Infrastructure Engineering*, vol. 15, no. 4, pp. 241–250, 2000. → [p]
- [98] J. P. Amezcuita-Sanchez and H. Adeli, "Synchrosqueezed wavelet transform-fractality model for locating, detecting, and quantifying damage in smart highrise building structures," *Smart Materials and Structures*, vol. 24, no. 6, p. 065034, 2015. → [p]
- [99] M. H. Rafiei, W. H. Khushefati, R. Demirboga, and H. Adeli, "Supervised deep restricted boltzmann machine for estimation of concrete," *ACI Materials Journal*, vol. 114, no. 2, p. 237, 2017. → [p28]
- [100] S.-N. Yu, J.-H. Jang, and C.-S. Han, "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel," *Automation in construction*, vol. 16, no. 3, pp. 255–261, 2007. → [p28]
- [101] J. Amezcuita-Sanchez and H. Adeli, "Feature extraction and classification techniques for health monitoring of structures," *Scientia Iranica*, vol. 22, no. 6, pp. 1931–1940, 2015. → [p28]
- [102] G. Sollazzo, K. C. Wang, G. Bosurgi, and J. Q. Li, "Hybrid procedure for automated detection of cracking with 3d pavement data," *Journal of Computing in Civil Engineering*, vol. 30, no. 6, p. 04016032, 2016. → [p28]
- [103] P. Prasanna, K. Dana, N. Gucunski, and B. Basily, "Computer-vision based crack detection and analysis," in *Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2012*, vol. 8345, pp. 1143–1148, SPIE, 2012. → [p29]

- [104] S. Chanda, G. Bu, H. Guan, J. Jo, U. Pal, Y.-C. Loo, and M. Blumenstein, "Automatic bridge crack detection—a texture analysis-based approach," in *Artificial Neural Networks in Pattern Recognition: 6th IAPR TC 3 International Workshop, ANNPR 2014, Montreal, QC, Canada, October 6-8, 2014. Proceedings 6*, pp. 193–203, Springer, 2014. → [p]
- [105] G. Bu, S. Chanda, H. Guan, J. Jo, M. Blumenstein, and Y. Loo, "Crack detection using a texture analysis-based technique for visual bridge inspection," *Electronic Journal of Structural Engineering*, vol. 14, no. 1, pp. 41–48, 2015. → [p]
- [106] R. Kapela, P. Śniatała, A. Turkot, A. Rybarczyk, A. Pożarycki, P. Rydzewski, M. Wyczalek, and A. Błoch, "Asphalt surfaced pavement cracks detection based on histograms of oriented gradients," in *2015 22nd international conference mixed design of integrated circuits & systems (MIXDES)*, pp. 579–584, IEEE, 2015. → [p29]
- [107] I. Abdel-Qader, S. Pashaie-Rad, O. Abudayyeh, and S. Yehia, "Pca-based algorithm for unsupervised bridge crack detection," *Advances in Engineering Software*, vol. 37, no. 12, pp. 771–778, 2006. → [p30]
- [108] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015. → [p32]
- [109] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3708–3712, IEEE, 2016. → [p32], [p33]
- [110] S. J. Schmugge, L. Rice, N. R. Nguyen, J. Lindberg, R. Grizzi, C. Joffe, and M. C. Shin, "Detection of cracks in nuclear power plant using spatial-temporal grouping of local patches," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–7, IEEE, 2016. → [p32], [p33]

- [111] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, “Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection,” *Construction and building materials*, vol. 157, pp. 322–330, 2017.
→ [p33]
- [112] K. Ma, M. Hoai, and D. Samaras, “Large-scale continual road inspection: Visual infrastructure assessment in the wild,” in *BMVC*, 2017.
→ [p]
- [113] K. Gopalakrishnan, H. Gholami, A. Vidyadharan, A. Choudhary, A. Agrawal, *et al.*, “Crack damage detection in unmanned aerial vehicle images of civil infrastructure using pre-trained deep learning model,” *Int. J. Traffic Transp. Eng.*, vol. 8, no. 1, pp. 1–14, 2018.
→ [p]
- [114] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui, and X. Chen, “Automatic bridge crack detection using a convolutional neural network,” *Applied Sciences*, vol. 9, no. 14, p. 2867, 2019.
→ [p33]
- [115] Y.-J. Cha, W. Choi, and O. Büyüköztürk, “Deep learning-based crack damage detection using convolutional neural networks,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
→ [p33]
- [116] F.-C. Chen, M. R. Jahanshahi, R.-T. Wu, and C. Joffe, “A texture-based video processing methodology using bayesian data fusion for autonomous crack detection on metallic surfaces,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 4, pp. 271–287, 2017.
→ [p]
- [117] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H.-M. Gross, “How to get pavement distress detection ready for deep learning? a systematic approach,” in *2017 international joint conference on neural networks (IJCNN)*, pp. 2039–2047, IEEE, 2017.
→ [p]

- [118] C. Feng, M.-Y. Liu, C.-C. Kao, and T.-Y. Lee, “Deep active learning for civil infrastructure defect detection and classification,” in *Computing in civil engineering 2017*, pp. 298–306, 2017. → [p]
- [119] L. Pauly, D. Hogg, R. Fuentes, and H. Peel, “Deeper networks for pavement crack detection,” in *Proceedings of the 34th ISARC*, pp. 479–485, IAARC, 2017. → [p]
- [120] K. C. Wang, A. Zhang, J. Q. Li, Y. Fei, C. Chen, and B. Li, “Deep learning for asphalt pavement cracking recognition using convolutional neural network,” in *Airfield and Highway Pavements 2017*, pp. 166–177, 2017. → [p]
- [121] X. Wang and Z. Hu, “Grid-based pavement crack analysis using deep learning,” in *2017 4th international conference on transportation information and safety (ICTIS)*, pp. 917–924, IEEE, 2017. → [p]
- [122] S. Yokoyama and T. Matsumoto, “Development of an automatic detector of cracks in concrete using machine learning,” *Procedia engineering*, vol. 171, pp. 1250–1255, 2017. → [p]
- [123] K. Chen, A. Yadav, A. Khan, Y. Meng, and K. Zhu, “Improved crack detection and recognition based on convolutional neural network,” *Modelling and simulation in engineering*, vol. 2019, pp. 1–8, 2019. → [p]
- [124] S. Li, X. Zhao, *et al.*, “Image-based concrete crack detection using convolutional neural network and exhaustive search technique,” *Advances in civil engineering*, vol. 2019, 2019. → [p]
- [125] S. Park, S. Bang, H. Kim, and H. Kim, “Patch-based crack detection in black box images using convolutional neural networks,” *Journal of Computing in Civil Engineering*, vol. 33, no. 3, p. 04019017, 2019. → [p33]

- [126] J. C. Cheng and M. Wang, “Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques,” *Automation in Construction*, vol. 95, pp. 155–171, 2018. → [p33]
- [127] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyaama, and H. Omata, “Road damage detection and classification using deep neural networks with smartphone images,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 12, pp. 1127–1141, 2018. → [p]
- [128] V. Mandal, L. Uong, and Y. Adu-Gyamfi, “Automated road crack detection using deep convolutional neural networks,” in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5212–5215, IEEE, 2018. → [p]
- [129] M. Nie and K. Wang, “Pavement distress detection based on transfer learning,” in *2018 5th International conference on systems and informatics (ICSAI)*, pp. 435–439, IEEE, 2018. → [p]
- [130] Y. Xue and Y. Li, “A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 8, pp. 638–654, 2018. → [p33]
- [131] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015. → [p33]
- [132] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 21–37, Springer, 2016. → [p33]
- [133] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014. → [p33], [p35]

- [134] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. → [p33]
- [135] M. D. Jenkins, T. A. Carr, M. I. Iglesias, T. Buggy, and G. Morison, “A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks,” in *2018 26th European signal processing conference (EUSIPCO)*, pp. 2120–2124, IEEE, 2018. → [p34]
- [136] J. Ji, L. Wu, Z. Chen, J. Yu, P. Lin, and S. Cheng, “Automated pixel-level surface crack detection using u-net,” in *Multi-disciplinary Trends in Artificial Intelligence: 12th International Conference, MIWAI 2018, Hanoi, Vietnam, November 18–20, 2018, Proceedings 12*, pp. 69–78, Springer, 2018. → [p]
- [137] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, “Deepcrack: Learning hierarchical convolutional features for crack detection,” *IEEE transactions on image processing*, vol. 28, no. 3, pp. 1498–1512, 2018. → [p34]
- [138] R. Ren, F. Liu, P. Shi, H. Wang, and Y. Huang, “Preprocessing of crack recognition: Automatic crack-location method based on deep learning,” *Journal of Materials in Civil Engineering*, vol. 35, no. 3, p. 04022452, 2023. → [p39]
- [139] X. Cui, Q. Wang, J. Dai, Y. Xue, and Y. Duan, “Intelligent crack detection based on attention mechanism in convolution neural network,” *Advances in Structural Engineering*, vol. 24, no. 9, pp. 1859–1868, 2021. → [p39]
- [140] Z. Qu, W. Chen, S.-Y. Wang, T.-M. Yi, and L. Liu, “A crack detection algorithm for concrete pavement based on attention mechanism and multi-features fusion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11710–11719, 2021. → [p]

- [141] X. Xiang, Y. Zhang, and A. El Saddik, "Pavement crack detection network based on pyramid structure and attention mechanism," *IET Image Processing*, vol. 14, no. 8, pp. 1580–1586, 2020. → [p]
- [142] C. Li, Y. Wen, Q. Shi, F. Yang, H. Ma, X. Tian, *et al.*, "A pavement crack detection method based on multiscale attention and hfs," *Computational intelligence and neuroscience*, vol. 2022, 2022. → [p39]
- [143] H. Liu, X. Miao, C. Mertz, C. Xu, and H. Kong, "Crackformer: Transformer network for fine-grained crack detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3783–3792, 2021. → [p40]
- [144] Z. Xu, H. Guan, J. Kang, X. Lei, L. Ma, Y. Yu, Y. Chen, and J. Li, "Pavement crack detection from ccd images with a locally enhanced transformer network," *International Journal of Applied Earth Observation and Geoinformation*, vol. 110, p. 102825, 2022. → [p]
- [145] E. Zhang, L. Shao, and Y. Wang, "Unifying transformer and convolution for dam crack detection," *Automation in Construction*, vol. 147, p. 104712, 2023. → [p]
- [146] S. Xiao, K. Shang, K. Lin, Q. Wu, H. Gu, and Z. Zhang, "Pavement crack detection with hybrid-window attentive vision transformers," *International Journal of Applied Earth Observation and Geoinformation*, vol. 116, p. 103172, 2023. → [p40]
- [147] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017. → [p43], [p55]
- [148] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017. → [p44], [p55]
- [149] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016. → [p45], [p55]

- [150] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph neural networks: A review of methods and applications,” *AI open*, vol. 1, pp. 57–81, 2020. → [p46]
- [151] E. Brissman, J. Johnander, M. Danelljan, and M. Felsberg, “Recurrent graph neural networks for video instance segmentation,” *International Journal of Computer Vision*, vol. 131, no. 2, pp. 471–495, 2023. → [p]
- [152] A. Aflalo, S. Bagon, T. Kashti, and Y. Eldar, “Deepcut: Unsupervised segmentation using graph neural networks clustering,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–41, 2023. → [p]
- [153] B. Zhang, J. Xiao, J. Jiao, Y. Wei, and Y. Zhao, “Affinity attention graph neural network for weakly supervised semantic segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8082–8096, 2021. → [p46]
- [154] S. Mohammadi and M. Allali, “Advancing brain tumor segmentation with spectral–spatial graph neural networks,” *Applied Sciences*, vol. 14, no. 8, p. 3424, 2024. → [p46]
- [155] J. Peng, L. Peng, Z. Zhou, X. Han, H. Xu, L. Lu, and W. Lv, “Multi-level fusion graph neural network: Application to pet and ct imaging for risk stratification of head and neck cancer,” *Biomedical Signal Processing and Control*, vol. 92, p. 106137, 2024. → [p]
- [156] J. Jiang, X. Chen, G. Tian, and Y. Liu, “Vig-unet: vision graph neural networks for medical image segmentation,” in *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, IEEE, 2023. → [p]

- [157] C. Saueressig, A. Berkley, E. Kang, R. Munbodh, and R. Singh, “Exploring graph-based neural networks for automatic brain tumor segmentation,” in *From Data to Models and Back: 9th International Symposium, DataMod 2020, Virtual Event, October 20, 2020, Revised Selected Papers 9*, pp. 18–37, Springer, 2021. → [p]
- [158] Z. Liu, C. Zhao, Y. Lu, Y. Jiang, and J. Yan, “Multi-scale graph learning for ovarian tumor segmentation from ct images,” *Neurocomputing*, vol. 512, pp. 398–407, 2022. → [p46]
- [159] C. Yang, I.-G. Todoran, and C. Saravia, “End-to-end road graph extraction based on graph neural network,” in *IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium*, pp. 4887–4890, IEEE, 2023. → [p47]
- [160] J. Chen, Y. Yuan, H. Lang, S. Ding, J. J. Lu, *et al.*, “The improvement of automated crack segmentation on concrete pavement with graph network,” *Journal of Advanced Transportation*, vol. 2022, 2022. → [p48], [p54]
- [161] H. Feng, L. Ma, Y. Yu, Y. Chen, and J. Li, “Scl-gcn: Stratified contrastive learning graph convolution network for pavement crack detection from mobile lidar point clouds,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 118, p. 103248, 2023. → [p48]
- [162] L. Berton, A. de Andrade Lopes, and D. A. Vega-Oliveros, “A comparison of graph construction methods for semi-supervised learning,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2018. → [p53]
- [163] W. Dong, C. Moses, and K. Li, “Efficient k-nearest neighbor graph construction for generic similarity measures,” in *Proceedings of the 20th international conference on World wide web*, pp. 577–586, 2011. → [p54]

- [164] X. Liu, M. Yan, L. Deng, G. Li, X. Ye, and D. Fan, “Sampling methods for efficient training of graph convolutional networks: A survey,” *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 2, pp. 205–234, 2022. → [p56]
- [165] N. Jaafar and Z. Lachiri, “Multimodal fusion methods with deep neural networks and meta-information for aggression detection in surveillance,” *Expert Systems with Applications*, vol. 211, p. 118523, 2023. → [p57]
- [166] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, “Dice loss for data-imbalanced nlp tasks,” *arXiv preprint arXiv:1911.02855*, 2019. → [p60]
- [167] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017. → [p61]
- [168] A. Mao, M. Mohri, and Y. Zhong, “Cross-entropy loss functions: Theoretical analysis and applications,” in *International Conference on Machine Learning*, pp. 23803–23828, PMLR, 2023. → [p61]
- [169] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017. → [p63]
- [170] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017. → [p64]
- [171] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, “Automatic road crack detection using random structured forests,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3434–3445, 2016. → [p65]

The road to success is always under construction.

– Lily Tomlin